IT UNIVERSITY OF COPENHAGEN

On Uncertainty In Natural Language Processing

DENNIS ULMER

Department of Computer Science
IT University of Copenhagen

Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

June 21, 2024

Committee

Advisor	Dr. Christian Hardmeier	IT Universitetet i København
Co-Advisor	Dr. Jes Frellsen	Danmarks Tekniske Universitet
Members	Dr. Leon Derczynski	IT Universitet i København
	Prof. Dr. Ole Winther	Danmarks Tekniske Universitet
	Prof. Dr. Mário A. T. Figueiredo	Instituto Superior Técnico

Abstract

The last decade in deep learning has brought on increasingly capable systems that are deployed on a wide variety of applications. In natural language processing, the field has been transformed by a number of breakthroughs including large language models, which are used in increasingly many user-facing applications. In order to reap the benefits of this technology and reduce potential harms, it is important to quantify the reliability of model predictions and the uncertainties that shroud their development.

This thesis studies how uncertainty in natural language processing can be characterized from a linguistic, statistical and neural perspective, and how it can be reduced and quantified through the design of the experimental pipeline. We further explore uncertainty quantification in modeling by theoretically and empirically investigating the effect of inductive model biases in text classification tasks. The corresponding experiments include data for three different languages (Danish, English and Finnish) and tasks as well as a large set of different uncertainty quantification approaches. Additionally, we propose a method for calibrated sampling in natural language generation based on non-exchangeable conformal prediction, which provides tighter token sets with better coverage of the actual continuation. Lastly, we develop an approach to quantify confidence in large black-box language models using auxiliary predictors, where the confidence is predicted from the input to and generated output text of the target model alone.

Resumé

Det sidste årti i deep learning har medført stadig mere dygtige systemer, der anvendes på mange forskellige områder. Feltet natural language processing (naturlig sprogbehandling) er blevet transformeret af en række gennembrud, herunder store sprogmodeller, som bruges i stadigt flere anvendelser med menneskelige brugere. For at udnytte fordelene ved denne teknologi og reducere potentielle skader, er det vigtigt at kvantificere pålideligheden af modelforudsigelser og de usikkerheder, der omkranser deres udvikling.

Dette afhandling undersøger, hvordan usikkerhed i natural language processing kan karakteriseres ud fra et sprogligt, statistisk og neuralt perspektiv, og hvordan den kan reduceres og kvantificeres gennem design af den eksperimentelle pipeline. Vi udforsker yderligere kvantificering af usikkerhed i modellering ved teoretisk og empirisk at undersøge effekten af modellers induktive bias i tekstklassificeringsopgaver. De tilsvarende eksperimenter omfatter data for tre forskellige sprog (dansk, engelsk og finsk) og opgaver samt et stort sæt forskellige tilgange til kvatificering af usikkerheder. Derudover foreslår vi en metode til kalibreret sampling i naturlig sproggenerering baseret på non-exchangeable conformal prediction, der giver smallere tokensæt med bedre dækning af den faktiske fortsættelse. Til sidst udvikler vi en tilgang til at kvantificere tillid i store black-box sprogmodeller ved hjælp af såkaldte hjælpeprædiktorer, hvor tilliden forudsiges ud fra input til og genereret outputtekst fra sprogmodellen alene.

Acknowledgements

Acknowledgements will be added in the print version of the thesis.

Declaration of Work

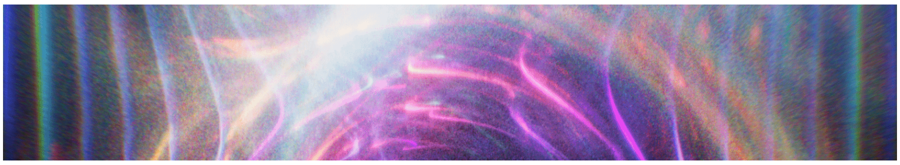
I, Dennis Ulmer, declare that this thesis—submitted in partial fulfillment of the requirements conferral of a PhD from the IT University of Copenhagen—is solely my own work unless otherwise referenced or attributed. Neither the thesis nor its content have been submitted (or published) for qualifications at another academic institution.

—Dennis Ulmer

Contents

List of Figures

List of Tables



Notation

> *"It is time for mathematicians, physicists, and computer scientists to forget their differences and admit that nobody really has a clue about what's going on in high dimensions."*
>
> —Clément Canonne

In the following, we generally follow the notational guidelines used in the book by Goodfellow et al. (2016) and by other organizations such as the Transactions on Machine Learning Research (TMLR) journal, with some modifications. These include the use of the following:

- Lowercase latin and greek letters for scalars, e.g. a, b, c and α, β, γ.
- Bold lowercase latin and greek letters for vectors, e.g. $\mathbf{a}, \mathbf{b}, \mathbf{c}$ and $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$.
- Bold uppercase latin and greek letters for matrices, e.g. $\mathbf{A}, \mathbf{W}$ and $\boldsymbol{\Theta}, \boldsymbol{\Psi}$.
- Uppercase letters such as $\mathbb{A}$ and $\mathbb{D}$ to denote sets. Sometimes, calligraphical letters like $\mathcal{C}$ might be used to denote sets when the notation might conflict with common conventions (e.g. $\mathbb{C}$ usually denoting the set of complex numbers.).
- $\{x_i\}_{i=1}^N$ to denote a set of elements $\{x_1, \ldots, x_N\}$. We also use the condensed shorthand $\{x_{ij}\}_{i,j=1}^{M,N}$ to denote a set of elements $\{x_{1,1}, \ldots, x_{M,1}, \ldots x_{M,N}\}$ indexed along two dimensions.
- $[K]$ to denote an set $\{1, 2, \ldots, K\}$, or more formally, for any $K \in \mathbb{N}^+$, $[K] = \{n \mid n \in \mathbb{N}^+ \text{ and } n \leq K\}$.

We denote an element-wise multiplication for vectors and matrices by $\circ$, and the same symbol may be used in some contexts to denote function compositions, i.e. $(f \circ g)(x) = g(f(x))$.

Definitions

Neural Network. Some concepts occur often enough to warrant their separate definitions. Since this thesis revolves around neural networks, we denote $\boldsymbol{\theta}$ as the (flattened) vector of network parameters and $\boldsymbol{\Theta}$ as the space of all possible weight parameters. Neural networks usually comprise a number of linear layers, consisting of a weight matrix $\mathbf{W}$ and a bias term $\mathbf{b}$, transforming inputs $\mathbf{x}$ into hidden encodings $\mathbf{z}$. A superscript or index might be added to indicate one of these objects belonging to a specific layer $l \in [L]$ or to a specific time step $t \in [T]$. Furthermore, we indicate with a index $\boldsymbol{\theta}$ when a function is parameterized by $\boldsymbol{\theta}$ (or some other set of parameters). This is generally done to reduce clutter and make equations more readable, but might be made explicit with conditioning when it is important in a statistical context. For instance, the probability distribution over classes $k \in [K]$ of a neural classifier will be denoted as $p_{\boldsymbol{\theta}}(y \mid \mathbf{x}) \equiv p(y \mid \mathbf{x}, \boldsymbol{\theta})$. In the same fashion, we denote $f_{\boldsymbol{\theta}}(\mathbf{x})$ as the logits, i.e. the unnormalized output of a neural classifier, and use $f_{\boldsymbol{\theta}}(\mathbf{x})_k$ to refer to the k-th logit.

Neural Network Functions. There also exist several functions that play a specific role in the neural network context. On of these is the sigmoid function defined as

$$\sigma(x) = \frac{1}{1 + \exp(-x)}, \tag{0.1}$$

as well as its multivariate generalization, the softmax function:

$$\text{softmax}(\mathbf{x})_k \equiv \bar{\sigma}(\mathbf{x})_k = \frac{\exp(x_k)}{\sum_{k=1}^{K} \exp x_k}, \tag{0.2}$$

where we sometimes will use the notation $\bar{\sigma}(\cdot)$ to avoid visual clutter.

Indicator function. The indicator function takes as input some condition, and evaluates as

$$\mathbb{1}\big(\text{condition}\big) = \begin{cases} 1 & \text{if condition is true} \\ 0 & \text{else} \end{cases} \tag{0.3}$$

In some cases it is useful to apply the indicator function elementwise to the contents of a vector. In that case, we use a bolded version, namely $\mathbf{1}(\cdot)$, which will be a vector of the same dimensionality. Take the example of a vector $\mathbf{x}$ whose elements are compared against a threshold τ. Then

$$\mathbf{1}(\mathbf{x} > \tau)_i = \begin{cases} 1 & \text{if } x_i > \tau \\ 0 & \text{else} \end{cases} \tag{0.4}$$

Statistics. In the context of statistics, we use the Dirac delta function, which is defined as 0 everywhere except for the origin, where it is $+\infty$:

$$\delta(x) = \begin{cases} +\infty & \text{if } x = 0 \\ 0 & \text{else} \end{cases} \tag{0.5}$$

In addition, its integral over the entire real number line is 1. Another set of definitions denotes common statistical concepts as the expectation of a random variable x

$$\mathbb{E}[x] = \sum_x P(x)x \quad \text{or} \quad \int_x p(x)\mathrm{d}x. \tag{0.6}$$

In this case, we also use P to denote probability mass functions and p to denote probability density functions. From this, we can also define the variance as

$$\mathrm{Var}[x] = \mathbb{E}\big[(x - \mathbb{E}[x])^2\big] \tag{0.7}$$

as well as the Shannon entropy

$$\mathrm{H}[x] = -\sum_x P(x)\log P(x) \quad \text{or} \quad -\int p(x)\log p(x)\mathrm{d}x. \tag{0.8}$$

Special Functions. Another set of definitions is dedicated to some mathematical functions, including the Gamma function $\Gamma(\cdot)$, which is a continuous version of the factorial and defined as

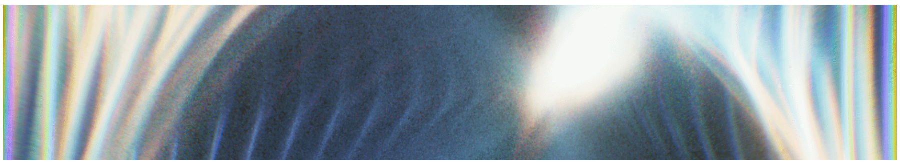
$$\Gamma(z) = \int_0^\infty t^{z-1}\exp(-t)\mathrm{d}t. \tag{0.9}$$

Another important function is the Beta function:

$$\mathbf{B}(\alpha_1, \alpha_2) = \frac{\Gamma(\alpha_1)\Gamma(\alpha_2)}{\Gamma(\alpha_1 + \alpha_2)}. \tag{0.10}$$

In some cases, we will consider the Beta function with an arbitrary number of α values. In that case, we collect them in a vector $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_K]^T$ and write the Beta function as

$$\mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{k=1}^K \Gamma(\alpha_k)}{\Gamma\Big(\sum_{k=1}^K \alpha_k\Big)}. \tag{0.11}$$



1 Introduction

"*Forudsigelse er meget vanskelig, især om fremtiden.*"

"*Prediction is very difficult, especially about the future.*"

—Niels Bohr

1.1 Motivation

Every person's life is full of decisions. Is this restaurant really as good as the reviews suggest? Should I take a job here or take a more interesting job in a city far away? These decisions can be hard to evaluate, since not all necessary information is known beforehand: Restaurant reviews might be fraudulent or biased, and a promising job opportunity might turn out to be different than advertised. Compare that with the example of making a move in a game of chess: Chess is called a game with *perfect* information, so all the positions and possible moves of the pieces on the board are known, and one could in theory make the optimal move at every step (assuming good chess-playing abilities). However, in real life we often do not have all the information necessary to make a perfect decision. As such, humans take into account the uncertainty that permeates their decision-making in order to manage risk.

In this way, machines are (or should be) no different. The decades-old research in machine learning (ML)—and especially the most recent advances in the last decade or so—have produced systems that make decisions from the mundane ("is this a picture of a cat or an airplane?") to the potentially risky ("what treatment should be recommended to this patient?"). This trend has been accelerated by the paradigm of *deep learning*, which allows us to build evermore complex systems that could solve increasingly complex tasks. The complexity of these systems through comes at the cost of losing a detailed understanding of all the "cogs and gears" involved due to the sheer size of models (including millions, billions and sometimes even trillions of such "gears"). This fact has spurred numerous lines of research to develop methods to make deep learning systems more robust, fair and safe.

One such line of research is concerned with *uncertainty quantification*, i.e. reflecting the degree of trustworthiness of a prediction. In systems with automatic decision-making, such scores can for instance be used to withhold a prediction or request human oversight. One popular example is autonomous driving: Consider an important traffic sign that cannot be accurately evaluated by the onboard computer, or a traffic situation that is hard to analyze. In these cases, a human driver might appreciate the opportunity to intervene with the car, e.g. by reducing its speed in the face of uncertainty, instead of the car sticking to a wrong assessment and endangering the driver's or other traffic users' lives.

At this point, the reader might be rightfully wonder whether such high-risk scenarios also exist for language applications. And indeed, such problems can arise in sometimes more, sometimes less obvious places. An intuitive application with these considerations is healthcare: More and more work has recently gone into building artificial intelligence (AI) systems that provide decision-support for medical staff. For instance, models could analyze text written by a user to detect signs of mental illness or triage (i.e., prioritize) patients when resources are limited (Cohan et al., 2016; Rozova et al., 2022; Stewart et al., 2022). In this case, uncertainty can serve as a signal to request an additional human review of a case. Confident but wrong predictions here can lead to a waste of resources, a loss of trust of the medical professionals in the system, and, in the worst case, leaving urgent cases untreated. As another example, natural language systems are also used to assist in legal deliberations (Chalkidis et al., 2019a; Martinez-Gil, 2023; Chalkidis, 2023). While the scenario of a "robo judge" is usually ruled out, there still remain risks where models used for legal discovery or research might overlook relevant or produce misleading or incorrect outputs.

Uncertainty quantification is an active research area for systems that operate for instance on images or tabular data, but it has only recently started to receive attention in the natural language processing (NLP) community. This thesis gives an introduction to uncertainty quantification in machine learning and natural language processing for novices, summarizes the current state of progress in the field, and presents some novel and relevant methods for some of the most pressing problems for automated languages processing: These include for instance determining the most viable methods in text classification and proposing new approaches to calibrated sampling for natural language generation, as well as confidence estimation for black-box models.

1.2 Applications

A lot of research on uncertainty quantification makes only superficial statements or tacit assumptions about its usefulness. The following, non-exhaustive list of aspects therefore underline potential practical use-cases.

Safety. In general, uncertainty estimates can improve safety whenever a system with automated decision capabilities could potentially have real-world effects. Some of these situations are studied in the AI safety literature (see e.g. Amodei et al., 2016): They can include preventing an intelligent agent from exploring unsafe options, or acting in a risky manner as its environment changes from the version it was trained with, which is often referred to as *distributional shift* (Shimodaira, 2000; Moreno-Torres et al., 2012). In these cases, uncertain options can either be outright rejected or decisions can be delegated to a human user.

Trust. In order to reap the benefits of automation and the ability to extract intricate patterns from large amounts of data, users have to trust the system's output, or otherwise run the danger of being mislead. In the worst case, they might grow to ignore or even antagonize an automatic system. Since our systems are inanimate—and often inscrutable—building trust between humans and machines can be a tricky endeavor. Nevertheless, there exists a notion of trust that can be built by consistency (i.e., knowing what to expect from a system) and by using uncertainty to understand the behavior of a model (Jacovi et al., 2021). We dedicate Section 2.4 to discuss this connection in more detail.

Fairness. A long line of works has demonstrated how modern deep learning systems have a tendency to discriminate against subpopulations in the dataset and how to mitigate these effects (see Caton and Haas, 2024; Mehrabi et al., 2021 for an overview). Additional studies have argued that this is the result of human biases in the machine learning pipeline (Waseem et al., 2021) as well as biases and underrepresentation of groups in the training dataset (Meng et al., 2022a). In the latter case, specific uncertainty quantification methods can indicate whenever the correct prediction is uncertain due to a lack of similar training data (see Sections 2.2.2 and 2.2.3). In other instances, *un*fairness might occur when models favor a prediction corresponding to a majority group in the dataset in the face of an inherently ambiguous input. Consider the example of machine translation system that is supposed to translate "*the doctor is here*" into Spanish. In English, we do not have to specify the gender of *doctor*, while this is necessary in Spanish. And thus, without any additional context, two translations are equally plausible ("*el doctor está aqui*" versus "*la doctora está aqui*"). Deep learning systems have an inclination to prefer the version that has appeared more often in the training data, which due to real-world human biases might be *el doctor* (Vanmassenhove et al., 2018). By exposing the inherent uncertainty however, we can delegate a series of decisions to the user or other specialized systems and avoid such pitfalls.

Efficiency. Not all inputs a deep learning system faces are equally difficult. Imagine a system that has been trained to distinguish images of lions and tigers. Upon receiving an picture of a lion similar to its training instances, we would expect a well-trained model to come to a confident (and correct) prediction. Many of our contemporary deep learning systems have grown to include from millions up to billions and sometimes trillions of learnable parameters, and thus incur considerable computational cost for every prediction. Therefore, some works have explored whether we can use notions of uncertainty to detect when a model has arrived at a secure prediction in order to skip unnecessary computations (i.e. Schuster et al., 2021, 2022). Conversely, consider that our fictional lion vs. tiger detector is faced with a liger, or an albino tiger displaying differently-colored fur.[1] In light of these difficult examples, we could use uncertainty to trigger additional computations to come to a conclusion (see an example for such a mechanism for machine translation by van der Poel et al., 2022). There is evidence that the human brain operators in a similar fashion, for instance when the reading time in human subjects increases when confronted with a surprising sentence structure (Ferreira and Henderson, 1991).

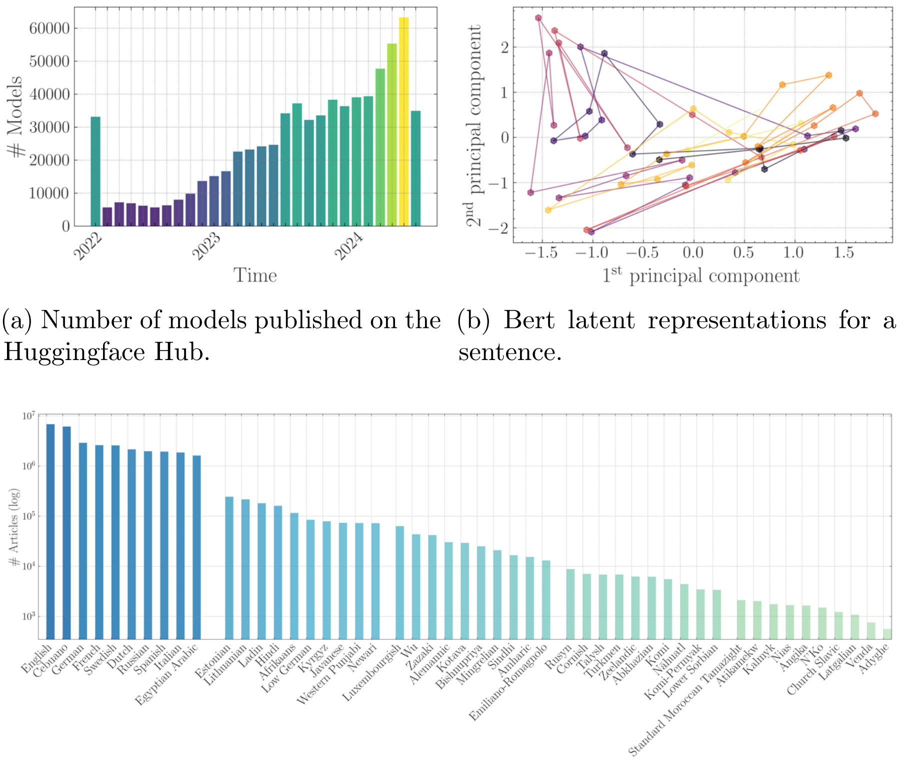
Interpretability. Due to the scale of modern architectures, the mechanisms in which they arrive at a prediction can be opaque and hard to deduce for humans. Here, research also has produced a variety of approaches to tackle this problem (see for instance Madsen et al., 2023 for a non-exhaustive selection). Uncertainty can be used as an additional angle to understand when the model might behave unreasonably confident or uncertain, with some studies already conducted for natural language generation (Ott et al., 2018; Xu et al., 2020; Xiao and Wang, 2021; Chen and Ji,

[1] A liger is a tiger / lion hybrid, see https://en.wikipedia.org/wiki/Liger.

2022).

Despite the variety of useful applications, there are a number of challenges to UQ that are very common or even unique in NLP, and distinguish this line of research from similar works on images or tabular data.

1.3 Challenges in Natural Language Processing



(a) Number of models published on the Huggingface Hub.

(b) Bert latent representations for a sentence.

(c) Number of Wikipedia articles by language (log-scale).

Figure 1.1: (a) Number of models published per month on the HuggingFace Hub (gathered on 17.06.2024). (b) Trajectory of the first two sentences of Turing (1950) using in the latent space of the uppermost layer of Bert (Devlin et al., 2019), after projecting them into two-dimensional space using PCA and whitening them. Time is indicated by color, reaching from dark (first token) to light (last token). (c) Number of articles by Wikipedia, log-scale (gathered on 11.04.2024). Shown are the top ten languages, and then ten randomly chosen languages of the remaining four quantiles of the distribution, each. All figures are best viewed in color and digitally.

The research in this thesis aims to fill a literature gap: While uncertainty quantification is a vibrant research field in machine learning, the availability of methods for natural language data is limited, and very few works develop solutions for this purpose specifically. This is disconcerting for the following reasons:

Challenges of Natural Language. In contrast to other machine learning problems, processing language is a rather messy affair. First of all, language is incredibly diverse, displaying vast differences between languages, dialects, demographics, domains or even individual speakers (Bender, 2011; Plank, 2016; Zampieri et al., 2020; van Esch et al., 2022). It is secondly embedded in a social and cultural context that is often necessary to understand its meaning (Hershcovich et al., 2022), and due to its paraphrastic nature, the idea same idea can often be expressed in a multitude of ways (Baan et al., 2023). Thirdly, the sequential nature often breaks the i.i.d. assumption that is a fundamental underlying assumption for many algorithms. One might assume that language data could just be treated as a time series and apply corresponding methods for uncertainty quantification (see e.g. Zhu and Laptev, 2017; Wang et al., 2020a; Blasco et al., 2024). Unfortunately though, encodings of language usually behave very erratically, as Figure 1.1b demonstrates. Modeling techniques for time series however subtly assume a certain behavior of the underlying data, e.g. a limit in the allowed rate of change encountered between two time steps.[2] The sometimes abrupt token-level changes encountered during language processing therefore prevent the application of time series modeling techniques.

Data Scarcity. Large amounts of both unstructured and annotated data exist for English, but most of the world's 7000+ languages are not blessed with such resources (Ruder, 2020; Joshi et al., 2020). Figure 1.1c shows the number of articles of a variety of Wikipedias on a log-scale. Due to its openness, Wikipedia remains a popular source of training data in NLP, however high-resource languages like English and German provide exponentially more potential training data compared to languages such as Afrikaans, Amharic or N'Ko.[3] This runs contrary to the strength of modern deep learning architectures: Weak architectural inductive biases such as in transformers enable us to learn complex meaning representations, but only when enough data is supplied (Tay et al., 2023). In the case of low-resource languages for instances, such data is often not available, and thus we can end

[2] This trait can for instance be formalized through the Lipschitz constant of the true data-generating function (Qu et al., 2022).

[3] Cebuano, the second-most spoken language in the Philippines, features the second-largest Wikipedia due to a bot called Lsjbot, which tries to create Cebuano Wikipedia articles for all living creatures. This makes around 99.6% of its articles bot-generated (Wikipedia contributors, 2024).

up with a model that is *underspecified* (D'Amour et al., 2022): The fewer training data points are available, the more possible models are able to fit them. While this might not lead to any problems on inputs similar to the training data, models might behave unpredictably on out-of-distribution data in ways that might not be immediately detectable by a user.

Trust & Safety. In machine learning, much of the research on uncertainty quantification is motivated by concerns regarding the trust in and the safety of automation. Despite this, similar research has until recently mostly remained nascent in NLP, despite being equally as relevant. Furthermore, the rapid developments in NLP with respect to large language models (Kalyan et al., 2021; Sevilla et al., 2022) have accelerated their adoption for a variety of applications by end users, albeit without appropriate techniques to ensure their safety. This trend is illustrated by Figure 1.1a, showing the number of models by month uploaded to the HuggingFace Hub, a platform to share open-source models. After a drop in submissions after its initial release in 2022, numbers have steadily increased to around $60k$ models in mid 2024. This, alongside a number of available proprietary models that can accessed through web interfaces and APIs, lowers the barrier of access to language models.

1.4 Objectives

This thesis analyzes the current state of uncertainty quantification research in deep learning and connects it to the methodological challenges that arise when they are applied to language data. As such, it aims to familiarize the reader with the most popular strategies for uncertainty quantification, as well as giving an intuition about their limitations. In this thesis, we seek to answer the following research questions:

🔎 **RQ1**: How can uncertainty in NLP be characterized?

Uncertainty can be a somewhat vague concept, and its definition is often passed over in different research works. Therefore, this thesis tries to gain a multi-disciplinary perspective on the matter, investigating different perspectives on the concept and how they are related.

🔎 **RQ2**: How can choices in experimental design help to reduce and quantify uncertainty?

Another overlooked factor in empirical research in NLP is the role of experimental design. Specifically, this work investigates how more conscious design decisions can not only help to reduce and quantify uncertainty, but also open new ways to model it.

🔎 **RQ3**: How do inductive model biases influence uncertainty quantification?

The *inductive biases* of a model usually refers to a set of implicit or explicit assumptions made by a learning algorithm (Hüllermeier et al., 2013). For instance, linear regression assumes that the target variable can be recovered as a linear combination of its input variables, and neural networks have an inductive bias to non-linear higher-order combinations of input features. The behavior of uncertainty in neural models is a priori often idealized (e.g. the model always displaying uncertainty on OOD), with this expectation sometimes being unfulfilled in practice. The reasons for this however are not so well understood, and thus this thesis sheds some light on the interaction of model biases and uncertainty.

🔎 **RQ4**: How can we address some of the challenges of uncertainty quantification in NLP?

Section 1.3 has listed some of the unique hurdles that UQ on NLP produces. Therefore, this thesis puts forth some new insights along with methodological advances to tackle these challenges.

1.5 Publications

The following works were produced during the PhD and are discussed in detail in this thesis (ordered chronologically). In all cases, the author's contributions amount to the main or complete share of the conception, implementation and description of ideas and experiments and the writing of the resulting publications, unless shared authorship is indicated.

1. **Ulmer, Dennis**[*], and Giovanni Cinà[*]. "Know your limits: Uncertainty Estimation with ReLU Classifiers Fails at Reliable OOD Detection." In: Uncertainty in Artificial Intelligence. PMLR (2021) (discussed in Section 4.1).

2. **Ulmer, Dennis**, Christian Hardmeier, and Jes Frellsen. "deep-significance-Easy and Meaningful Statistical Significance Testing in the Age of Neural Networks." In: The

[*] Equal Contribution.

Workshop on Machine Learning Evaluation Standards at ICLR (2022) (discussed in Section 3.2).

3. **Ulmer, Dennis**, Elisa Bassignana, Max Müller-Eberstein, Daniel Varab, Mike Zhang, Rob van der Goot, Christian Hardmeier, and Barbara Plank. "Experimental Standards for Deep Learning in Natural Language Processing Research." In: Findings of the Association for Computational Linguistics: EMNLP (2022), pp. 2673–2692 (discussed in Section 3.1).

4. **Ulmer, Dennis**, Jes Frellsen, and Christian Hardmeier. "Exploring Predictive Uncertainty and Calibration in NLP: A Study on the Impact of Method & Data Scarcity." In: Findings of the Association for Computational Linguistics: EMNLP (2022), pp. 2707—2735 (discussed in Section 4.2).

5. **Ulmer, Dennis**, Christian Hardmeier, and Jes Frellsen. "Prior and posterior networks: A survey on evidential deep learning methods for uncertainty estimation." In: Transactions on Machine Learning Research. JMLR (2023) (discussed in Section 2.2.3).

6. **Ulmer, Dennis**, Chrysoula Zerva, André FT Martins: "Non-Exchangeable Conformal Language Generation with Nearest Neighbors". In: Findings of the Association for Computational Linguistics: EACL (2024), pp. 1909–1929 (discussed in Chapter 5).

7. **Ulmer, Dennis**, Martin Gubri, Hwaran Lee, Sangdoo Yun, Seong Joon Oh. "Calibrating Large Language Models Using Their Generations Only". In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (discussed in Chapter 6).

The following works were produced during the PhD, but will not be discussed in detail, either since the author was not the main author, or because they were not a good fit for the topic of this thesis:

8. Baan, Joris*, Nico Daheim*, Evgenia Ilia*, **Dennis Ulmer***, Haau-Sing Li, Raquel Fernández, Barbara Plank, Rico Sennrich, Chrysoula Zerva, Wilker Aziz. "Uncertainty in Natural Language Generation: From Theory to Applications." Under review, 2024.

9. Hupkes, Dieuwke, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella

Sinclair, **Dennis Ulmer**, Florian Schottmann, Khuyagbaatar Batsuren, Kaiser Sun, Koustuv Sinha, Leila Khalatbari, Rita Frieske, Ryan Cotterell, Zhijing Jin: “A Taxonomy and Review of Generalization Research in NLP”. In: Nature Machine Intelligence 5 (10), p. 1161–1174.

10. Farinhas, António, Chrysoula Zerva, **Dennis Ulmer**, André FT Martins. “Non-exchangeable Conformal Risk Control”. In: Proceedings of the International Conference on Learning Representations, 2024.

11. **Ulmer, Dennis**, Elman Mansimov, Kaixiang Lin, Justin Sun, Xibin Gao, Yi Zhang. “Bootstrapping LLM-based Task-Oriented Dialogue Agents via Self-Talk”. In: Findings of the Association for Computational Linguistics: ACL 2024.

12. Gubri, Martin, **Dennis Ulmer**, Hwaran Lee, Sangdoo Yun, Seong Joon Oh. “TRAP: Targeted Random Adversarial Prompt Honeypot for Black-Box Identification”. In: Findings of the Association for Computational Linguistics: ACL 2024.

Another published document concerns the proceedings of the UncertaiNLP workshop, in which the author was involved as an editor and workshop co-organizer. The workshop was co-located with the European meeting of the Association for Computational Linguistics (EACL) in St. Julians, Malta, in 2024:

- Vázquez, Raúl, Hande Celikkanat, **Dennis Ulmer**, Jörg Tiedemann, Swabha Swayamdipta, Wilker Aziz, Barbara Plank, Joris Baan, and Marie-Catherine de Marneffe. Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024).

All of the above publications are accompanied by open-source code, that is listed in detail in Appendix C.1. However, some of the more important open-source contributions are highlighted here:

1. `nlp-uncertainty-zoo`: A Python package implementing different methods for uncertainty quantification in sequence classificationand sequence labeling in NLP.

2. `deep-significance`: A Python package including many functions to simplify statistical significance testing in deep learning.

3. `awesome-experimental-standards-deep-learning`: A pointer to useful resources in order to improve experimental standards as well as reproducibility and replicability in Deep Learning experiments.

1.6 Structure

This thesis is structured as to provide a comprehensive overview over the topic of uncertainty from both a statistical and linguistic point of view. Both perspectives are then woven together in a overview over uncertainty quantification in deep learning and natural language processing. This part serves as a foundation for later chapters about the uncertainty in the experimental design in NLP, before concretely tackling specific problem scenarios: Uncertainty in text classification problems, uncertainty in language generation problems and uncertainty in latter problems specifically involving the use of large language models.

To be more detailed, Chapter 2 introduces the reader to different concepts in uncertainty quantification and related literature. It begins with a definition of uncertainty from a variety of perspectives, for instance frequentist and Bayesian statistics, linguistics, and several popular approaches in deep learning. In this context, we also discuss Ulmer et al. (2023), which surveys works related to a novel class of uncertainty quantification methods called *evidential deep learning*. In the end, this includes a discussion of the relationship of uncertainty quantification with the end-user with both a motivation in trust and communication.

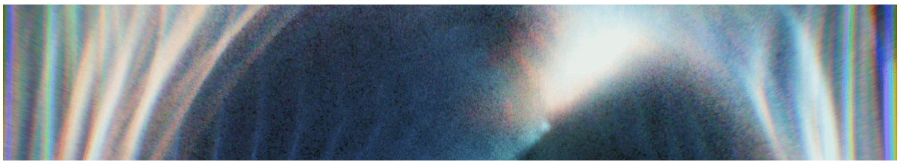
While most of the research that makes up this thesis is focused on uncertainty in *modeling* language, uncertainty also occurs in the experimental design and execution of day-to-day research. Therefore, Chapter 3 presents an interlude on challenges with the notions of reproducibility & replicability in deep learning, their connection to uncertainty, and the use of statistical hypothesis testing, all of which inform the methodology of later chapters. This encompasses the published works of Ulmer et al. (2022a), giving an account of ongoing discussions about experimental methods in deep learning, as well as Ulmer et al. (2022c), introducing a package for better statistical hypothesis testing and its application to a case study with large language models.

In the subsequent Chapter 4, we tackle the problem of uncertainty in classification problems. First we demonstrate the pitfalls of uncertainty quanitification for classification using simple ReLU networks, drawing from Ulmer and Cinà (2021). Afterwards, we discuss uncertainty quantification in the context of different classification problems specific to NLP, based on Ulmer et al. (2022b). Here we show how well exisiting methods for NLP fare on different languages and tasks, and how much that performance—including the reliability of uncertainty estimates—depends on the amount of available training data.

In Chapter 5, we move to the problem of natural language generation, and develop a new calibrated sampling method based on conformal prediction (Papadopoulos et al., 2002; Vovk et al., 2005). In language generation, we often restrict the set of possible candidate tokens to generate to a subset of (hopefully) plausible continutations. Using the theoretical underpinning of conformal prediction, we introduce a novel method that does so with statistical guarantees. This chapter is based on Ulmer et al. (2024c), and we demonstrate how this novel way to construct prediction sets is theoretically sound and produces flexibel prediction sets that come with guarantees about containing plausible tokens to generate.

In Chapter 6, we discuss a new method for quantifying the confidence of LLMs originally published in Ulmer et al. (2024a), that tries to circumvent many of the pratical constraints that come with large model sizes. Compared to other alternatives, this method is furthermore applicable to black-box models that do not allow any internal access to model states or weights, and is comparatively lightweight to train.

The thesis continues with a general discussion of the overall results in Chapter 7. There, we answer the overarching research questions stated in Section 1.4 and reflect on current research directions in the field. Lastly, Chapter 8 takes on a bird's-eye view by contextualizing uncertainty quantification in the current zeitgeist and discussing its relationship with contemporary policies. In addition, the thesis comprises an appendix with theoretical results (Appendix A) and one with experimental details (Appendix B) that were omitted from the main text. Details that are necessary for an accurate reproduction of experimental results are bundled in Appendix C.



2 | Background

"Le doute n'est pas une état bien agréable, mais l'assurance est un état ridicule."

"Doubt is not a pleasant condition, but certainty is absurd."

—Voltaire

Uncertainty is a common occurrence in everyone's life, and thus most people have an intuitive understanding of the concept. To define it concretely, however, can be challenging. Colloquially, we might define uncertainty as a phenomenon or state that is filled with doubts, lack knowledge or that is simply hard to predict. In research papers, the term uncertainty often only remains vaguely defined, either building on an intuitive definition or presupposing a certain school of thinking.

The aim of this chapter is to bring some clarity to the different ways uncertainty is defined, and to give an fairly comprehensive account of its applications. This entails a journey from its origins in statistics (Sections 2.1.1 and 2.1.2) and linguistics (Sections 2.1.3 and 2.1.4) to its implementation with neural networks, specifically in deep learning (Section 2.2) and natural language processing (Section 2.3). In the latter contexts, this comes with a focus on *modeling* uncertainty, and this is indeed also where many of the research papers in the field end. Therefore, an additional goal of this chapter is to not take uncertainty modeling as the ultimate goal per se, but to see beyond it and grasp the bigger picture. As uncertainty quantification is often motivated by increasing trustworthiness and safety, we take a closer look at the relationship between uncertainty and trust in Section 2.4, as well as how to communicate uncertainty in Section 2.5. Furthermore, the chapter outlines diverse applications of uncertainty in Section 2.6.

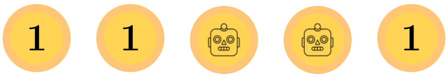
2.1 What Is Uncertainty, anyway?

We start by defining the most central concept in this thesis: Uncertainty. Since this thesis is focused on NLP, we aim to define the concept from all the perspectives modern NLP touches on. This includes building up some basic concepts from frequentist and Bayesian statistics as well from different parts of linguistics.

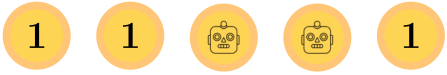
2.1.1 The Frequentist Perspective

> "*Statistical inference is serious business.*"
>
> —Bradley Efron, Robert J. Tibshirani in *An Introduction to the Bootstrap* (Tibshirani and Efron, 1993)

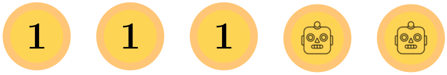
Frequentist statistics in an approach to statistics that aims to make inferences and draw conclusions from sampled data, alone. The term is based on the fact that probabilities are seen as equivalent to the observed frequencies of events in the data, assuming (potentially infinitely) many repetitions of an experiment (Willink and White, 2011). Let us reason about the popular example of a coin flip here to illustrate this notion. We are given a coin and would like to estimate the probability of heads, which we define as the parameter of interest to estimate and will denote by θ. We do not know whether the coin is fair, so we flip it a number of times and count the heads and tails to estimate this probability. We obtain the following five coin flips:



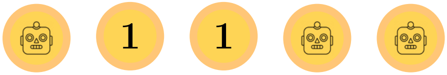
Based on this experiment, we then estimate the probability of heads as $\hat{\theta} = \frac{\#\text{heads}}{\#\text{coin flips}} = \frac{2}{5} = 0.4$. However, how can we be sure that this reflects the actual probability of heads? We thus repeat the experiment three more times, and obtain:



$\rightarrow \hat{\theta}_2 = \frac{2}{5} = 0.4$



$\rightarrow \hat{\theta}_3 = \frac{2}{5} = 0.4$



$\rightarrow \hat{\theta}_4 = \frac{3}{5} = 0.6$

As we gather more and more samples and take their average, we will provably converge to the true value of θ in the limit due to the law of large numbers (Dekking et al., 2005). But in light of a limited number of samples like above, how can we quantify the uncertainty of our estimate?

Confidence Intervals. One common approach to compute some frequentist uncertainty estimate is the use of *confidence intervals* (Neyman, 1937). Confidence intervals try to capture a range of values for the parameter θ such that, if we were to repeat our experiment, the computed confidence intervals would cover the true value with some probability (e.g. 95%). We can do this by assuming that any estimates $\hat{\theta}$ for θ are independently and normally distributed. The normality assumption holds in this case since the central limit theorem applies, stating that if we were to repeat this experiment over and over, the sample mean (which is our estimate $\hat{\theta}$) will be normally distributed. Possessing the knowledge about the estimate being normally distributed lets us define the confidence intervals. The procedure is as follows: We know that our samples are normally distributed according to some specific (constant but unknown) mean θ and standard deviation. In our example, θ would correspond to the true probability of heads that we are interested in. For convenience, we now standardize this distribution. We can achieve this by simply subtracting the mean θ from the mean of our estimates $\bar{\theta}$ and dividing by an estimate of the standard deviation denoted as s, which we obtain as:

$$\bar{\theta} = \frac{1}{N}\sum_{i=1}^{N}\hat{\theta}_i \tag{2.1}$$

$$s^2 = \frac{1}{N-1}\sum_{i=1}^{N}(\hat{\theta}_i - \bar{\theta})^2. \tag{2.2}$$

According to the central limit theorem, the estimate of the standard deviation improves in accuracy by a factor of $\frac{1}{\sqrt{N}}$, leading to the standard error $s/\sqrt{N}$, and we thus arrive at:

$$t = \frac{\bar{\theta} - \theta}{s/\sqrt{N}}. \tag{2.3}$$

Now, we would like to know how the statistic in Equation (2.3) is distributed in order to identify confidence intervals for θ. We already know that $\bar{\theta}$ is distributed according to a Normal distribution, and assuming that the sample variance s^2 is distributed according to a χ^2 distribution with N degrees of freedom, we obtain a Student's-t distribution with $N-1$ degrees of freedom. We can now determine the bounds such that $p(-c \leq t \leq c) = 0.95$. Using some intermediate steps, we find that

$$\begin{aligned}
&p(-c \leq t \leq c) && (2.4)\\
=&p\Big(-c \leq \frac{\bar{\theta}-\theta}{s/\sqrt{N}} \leq c\Big) && (2.5)\\
=&p\Big(-\frac{cs}{\sqrt{N}} \leq \bar{\theta}-\theta \leq \frac{cs}{\sqrt{N}}\Big) && (2.6)\\
=&p\Big(-\frac{cs}{\sqrt{N}}-\bar{\theta} \leq -\theta \leq \frac{cs}{\sqrt{N}}-\bar{\theta}\Big) && (2.7)\\
=&p\Big(\bar{\theta}-\frac{cs}{\sqrt{N}} \leq \theta \leq \bar{\theta}+\frac{cs}{\sqrt{N}}\Big). && (2.8)
\end{aligned}$$

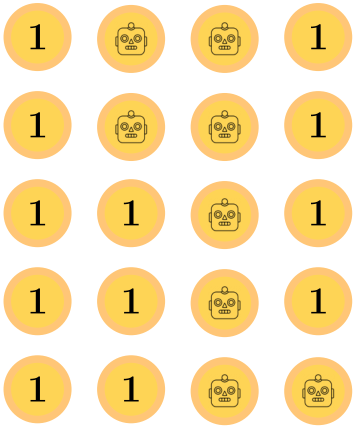
Therefore, we know that our unknown mean θ of the distribution will be contained within these bounds. Lastly, we need to choose c such that the proposed interval corresponds to 95% (or some other desired amount) of the total probability density. Since the shape of the Student's-t distribution is known, we can choose c to correspond to the 97.5-th percentile (leaving 2.5% of the total density to either side). This number can be easily computed through the distributions's inverse cumulative distribution function.[4] In our example above, we have $N = 4$, $\bar{\theta} = \frac{1}{4}(\frac{2}{5}+\frac{2}{5}+\frac{2}{5}+\frac{3}{5}) = 0.45$ and $s^2 = \frac{1}{3}(0.05^2+0.05^2+0.05^2+0.15^2) \approx 0.0111$. With $c \approx 3.182$, this gives us a confidence interval of $[0.31, 0.59]$. This interval can be interpreted in the following way: If we were to repeat this experiment, say, 100 times, the resulting confidence intervals would contain the true value 95 times. For the samples shown above a fair coin was used, and thus the confidence interval contains the true value of 0.5. It should be noted here that confidence intervals are sometimes available in closed form, for instance for the normal distribution.[5] This simplistic example also assumed confidence intervals to be both symmetrical and two-sided (i.e. having a lower and upper bound). For a more thorough and comprehensive treatment of confidence intervals we refer to other works such as Zech (2002); Smithson (2003).

Bootstrapping and Jackknife. In the previous example, we were able to successfully obtain a confidence interval for the

[4] Using a software library such as `scipy`, we can for instance compute `scipy.stats.t.ppf(0.975, df=3)`.

[5] The reasoning goes as follows: By Cochran's theorem, if the distribution is normal, the sample mean and variance are independent (Cochran, 1934). But the reverse is also true, so with an independent sample mean and variance we can assume that the underlying distribution is normal. We can again construct the t-statistic in Equation (2.3) to obtain confidence intervals e which are $\theta \in [\bar{\theta} - t_{n-1,1-\alpha/2}\frac{s}{\sqrt{N}}, \bar{\theta} + t_{n-1,1-\alpha/2}\frac{s}{\sqrt{N}}]$, where $t_{N-1,1-\alpha/2}$ stands for the $1-\alpha/2$-th quantile of a t-distribution with $N-1$ degrees of freedom and a $1-\alpha$ confidence level (Krishnamoorthy, 2006, p. 130).

probability of heads. However, this required us to repeat the coin flipping experiment multiple times. In the case of flipping a coin, this is rather straightforward—nevertheless, we might also interested in quantifying the uncertainty about the estimate in cases where obtaining a new sample is difficult or expensive (e.g. when an experiment require expensive computational hardware). A tool for these cases is given in the form of *bootstrapping* (Efron, 1992; Tibshirani and Efron, 1993): Instead of collecting new data, we can instead perform inferences from our existing sample through re-sampling: We sample randomly from our initial set of coin flips with replacement,[6] and obtain a number of new (pseudo-)samples. We then use these to estimate the confidence intervals of our estimate in a similar fashion to the confidence intervals in the previous paragraph: Drawing $N = 10$ re-samples, using the same procedure as in Equations (2.1) and (2.4), we obtain a confidence interval of $[0.26, 0.62]$. Nevertheless, there are known problems with the bootstrap: When our sample size small (just five coin flips), the sample might not be representative, and bootstrap samples can amplify any bias present in the sample. Here, having two heads and three tails differs slightly from the actual probability of 0.5, which is then carried over into the bootstrap samples. Another, similar estimator is the *jackknife* (Quenouille, 1949; Tukey, 1958), where we do not resample, but instead create new samples by leaving out one observation at a time. Therefore, the original set of coin flips would yield the following new pseudo-samples:



$$\rightarrow \hat{\theta}_{-1} = \tfrac{2}{4} = 0.5$$

$$\rightarrow \hat{\theta}_{-2} = \tfrac{2}{4} = 0.5$$

$$\rightarrow \hat{\theta}_{-3} = \tfrac{1}{4} = 0.25$$

$$\rightarrow \hat{\theta}_{-4} = \tfrac{1}{4} = 0.25$$

$$\rightarrow \hat{\theta}_{-5} = \tfrac{2}{4} = 0.5$$

We again repeat our procedure in Equations (2.1) and (2.4) to obtain the confidence interval of $[0.25, 0.55]$. In order to make a prediction about a new coin flip, in all of three cases we would

[6] Sampling with replacement implies that our new samples might contain duplicates.

simply declare head with a probability of the estimated $\hat{\theta}$ or hedge our bets using the range of values contained in the confidence interval.

Likelihood Functions. A useful tool to evaluate the fit of a parameter estimate for the data are *likelihood functions*. The likelihood $p(\mathbb{D} \mid \theta)$ quantifies how well the choice of a value for θ "explains" the observations $\mathbb{D}$, meaning how likely the value is to have generated the data or how consistent the data are with the chosen value. Accordingly, a high likelihood expresses that a value of θ is consistent with the observations, while a low likelihood suggest that θ is unlikely to have generated the data. For the coin flipping example, we can choose a *Bernoulli* likelihood:

$$\text{Bernoulli}(x \mid \theta) = \theta^x(1-\theta)^{(1-x)}. \tag{2.9}$$

Given the probability of heads θ, it assigns a probability of an outcome x, i.e. heads or tails. In line with the intuition that more suitable values of θ assign higher likelihoods to the data, a quick derivation reveals that the mean $\hat{\theta}$ we used is actually the parameter that maximizes the likelihood of our sample:

$$\begin{aligned}
p(\mathbb{D} \mid \theta) &= \prod_{i=1}^{N} p(x_i \mid \theta) = \prod_{i=1}^{N} \theta^x(1-\theta)^{(1-x)} && (2.10)\\
\log p(\mathbb{D} \mid \theta) &= \sum_{i=1}^{N} x_i \log(\theta) + (1-x_i)\log(1-\theta) && (2.11)\\
\frac{\partial}{\partial\theta}\log p(\mathbb{D} \mid \theta) &= \sum_{i=1}^{N} \frac{x_i}{\theta} - \frac{1-x_i}{1-\theta} \overset{!}{=} 0 && (2.12)\\
0 &= \sum_{i=1}^{N} \frac{x_i(1-\theta)}{\theta(1-\theta)} - \frac{(1-x_i)\theta}{\theta(1-\theta)} && (2.13)\\
&= \sum_{i=1}^{N} x_i - \cancel{x_i\theta} - \theta + \cancel{x_i\theta} && (2.14)\\
\theta_{\text{MLE}} &= \frac{1}{N}\sum_{i=1} x_i. && (2.15)
\end{aligned}$$

Here, we used the i.i.d. assumption (identically, independently distributed) to argue that since observations x_i are independent, we can factor the joint distribution $p(\mathbb{D} \mid \theta) \equiv p(x_1, \ldots, x_N \mid \theta)$ into a product of its individual likelihoods. Then, we transfer the computation into log-space for convenience, and identify the estimate by taking the derivative w.r.t. θ, setting to zero, and solving for it. This is referred to as the *maximum likelihood estimate* (MLE).

Recap. The above methods have illustrated the viewpoint of frequentist statistics: Parameter estimates are derived from the actually observed data, and our uncertainty about the estimates can expressed through confidence intervals, in which we expect our true value to fall. These can be obtained by relating estimates of the parameter for instance to the Student's-t distribution by exploiting the central limit theorem. Furthermore, other estimates can be obtained by collecting more data or through procedures such as the bootstrap or the jackknife. In our case, we knew that the used coin was fair, and that the initial sample simply ended up not representative due to using an uneven number of observations. In a similar but more realistic scenario, we might not know anything about the properties of the coin (or the phenomenon of interest), but might still suspect, at least without any other information available, that it is fair. The frequentist framework does not give us any means to incorporate this belief into our reasoning, but the Bayesian view presented in the next section does.

2.1.2 The Bayesian Perspective

> "*There is a valid defence of using non-Bayesian methods, namely incompetence.*"
>
> —John Skilling in *Fundamentals of MaxEnt in Data Analysis* (Skilling and Sibisi, 1990).

Bayesian statistics delineates itself from frequentist statistics by seeing probability itself as more than just the mere relative frequency of an event, and instead as the degree of belief in the occurrence of an event.[7] This difference has caused (and is still causing) ideological chasms among statisticians, as illustrated by the quote above. The name of Bayesian statistics is derived from Thomas Bayes, an English presbytarian minister in the 18th century who first formulated the eponymous *Bayes' theorem*. It should be noted however that Bayes only formulated his theory in a very specific setting,[8] and that a general version of Bayesian statistics was instead pioneered by Pierre-Simon Laplace (McGrayne, 2011; Leonard, 2014). The theorem can be formulated as follows: Given a

[7] Even though there are also subtle nuances to this definition, see for instance Good (1971).

[8] Namely, using a uniform prior. See Equation (2.16) and onward.

set of observations $\mathbb{D}$ and a parameter of interest θ, we can express the probability of the parameter given the observational data as

$$p(\theta \mid \mathbb{D}) = \frac{p(\mathbb{D} \mid \theta)p(\theta)}{p(\mathbb{D})}, \tag{2.16}$$

where the different parts of the equation are commonly referred to as the *posterior* $p(\theta \mid \mathbb{D})$, the likelihood $p(\mathbb{D} \mid \theta)$, the prior $p(\theta)$, and the evidence $p(\mathbb{D})$. We already discussed likelihoods in the previous section. The prior $p(\theta)$ is a probability distribution over possible values of θ, and thus allows us to express our prior belief by attributing higher probability to values of θ we deem more likely. This also implies a philosophical difference with frequentist statistics: While θ was treated as an unknown constant before, it is now seen as another random variable. The evidence $p(\mathbb{D})$ encodes the general probability of the observed data under any value of θ. This somewhat hidden interpretation becomes more clear when rewriting the term:

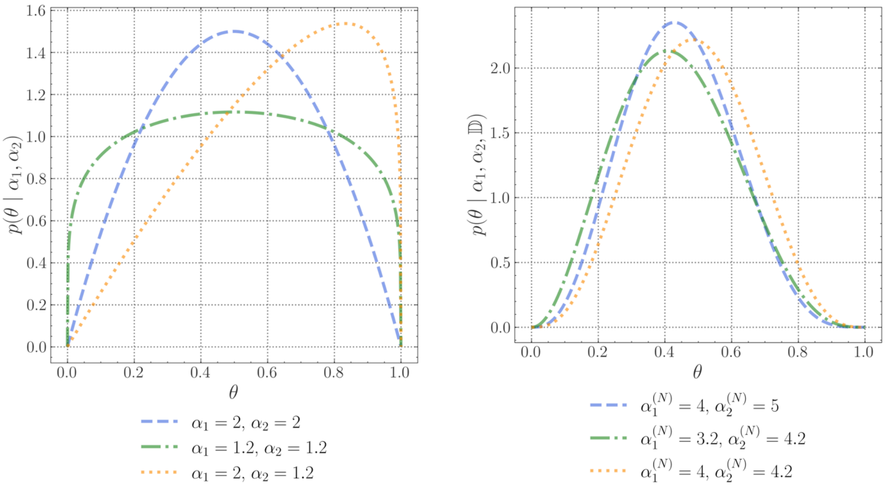
$$p(\mathbb{D}) = \int p(\mathbb{D}, \theta)\mathrm{d}\theta = \int p(\mathbb{D} \mid \theta)p(\theta)\mathrm{d}\theta. \tag{2.17}$$

We can therefore interpret the evidence as the likelihood of the data averaged over all possible parameter values of θ, weighed by their prior probabilities. Lastly, the posterior $p(\theta \mid \mathbb{D})$ describes a probability distribution over values of θ given our observations. We can think of the posterior as starting with our prior belief, using the data to update it and arriving at a final distribution that takes both of these into account. This has several advantages: We can now choose to encode our suspicions about the value of the target parameter into the prior. But as we will see, obtaining more and more data points results in outweighing the prior belief, completely relying on the observations in the limit.

Coin Flipping Redux. We now illustrate these concepts using the coin flipping example from Section 2.1.1, showing how uncertainty is modeled from the Bayesian perspective. In order to do so, we first have to make some design choices, i.e. the choice of likelihood and prior function as well as prior parameters. We again use the Bernoulli likelihood from the previous section, and now would like to define a prior over θ. A good choice for a prior for the Bernoulli distribution is the *Beta* distribution:

$$\text{Beta}(\theta; \alpha_1, \alpha_2) = \frac{1}{\mathrm{B}(\alpha_1, \alpha_2)}\theta^{\alpha_1 - 1}(1-\theta)^{\alpha_2 - 1} \tag{2.18}$$

$$\mathrm{B}(\alpha_1, \alpha_2) = \frac{\Gamma(\alpha_1)\Gamma(\alpha_2)}{\Gamma(\alpha_1 + \alpha_2)}, \tag{2.19}$$



(a) Beta priors.

(b) Beta posteriors.

Figure 2.1: Plots of different choices of (a) Beta prior distributions and their resulting (b) posterior distributions after observing our initial set of coin flips. Juxtaposing the two plots illustrates how the choice of prior belief can influence the shape of the resulting posterior distribution.

with $\Gamma(\cdot)$ denoting the Gamma function, a generalization of the factorial to the real numbers. The distribution has its support on $[0, 1]$ and possesses two shape parameters $\alpha_1, \alpha_2 \in \mathbb{R}^+$, which we can use to encode our prior belief about θ. A few examples for the resulting distribution are shown in Figure 2.1a. Choosing the Beta distribution as a prior comes in handy when analytically deriving the posterior distribution, since it is *conjugate* to the likelihood. Conjugacy here means that using a Beta prior together with a Bernoulli likelihood as in Equation (2.10), the posterior has the form of a Beta distribution.[9] Bayes' rule contains the unwieldy evidence term, which we established in Equation (2.17) can in some cases be evaluated analytically using an integral over parameters. However, we can notice that the evidence $p(\mathbb{D})$ does not depend on the parameters directly, and only scales the posterior $p(\theta \mid \mathbb{D})$ by a constant. As such, we can declare the posterior to be proportional to the product of the likelihood and prior:

$$p(\theta \mid \mathbb{D}) = \frac{p(\mathbb{D} \mid \theta)p(\theta)}{p(\mathbb{D})} \propto p(\mathbb{D} \mid \theta)p(\theta) = \prod_{i=1}^{N} p(x_i \mid \theta)p(\theta), \quad (2.20)$$

[9] Conjugate priors are available for distributions that can be generalized to a particular form which is referred to as *exponential families*, including popular distributions such as the Normal, Poisson, Bernoulli, and categorical distribution and more (Bishop and Nasrabadi, 2006; Gelman et al., 2021; Efron, 2022).

which simplifies solving for the posterior parameters. We now substitute the expressions in Equations (2.9) and (2.18) into Bayes' rule in Equation (2.16) and continue in log-space for convenience:

$$
\begin{aligned}
\log p(\theta \mid \mathbb{D}) \propto & \sum_{i=1}^{N} \log p(x_i \mid \theta) + \log p(\theta) && (2.21)\\
= & \sum_{i=1}^{N} x_i \log \theta + (1 - x_i) \log(1 - \theta) \\
& + (\alpha_1 - 1) \log \theta + (\alpha_2 - 1) \log(1 - \theta) && (2.22)\\
= & \Big(\alpha_1 + \sum_{i=1}^{N} x_i - 1\Big) \log \theta \\
& + \Big(\alpha_2 + N - \sum_{i=1}^{N} x_i - 1\Big) \log(1 - \theta), && (2.23)
\end{aligned}
$$

where we can see that in the end—after dropping the log-Beta function as it is just a constant—we obtain the form of a Beta distribution, but this time with the new shape parameters $\alpha_1^{(N)} = \alpha_1 + \sum_{i=1}^{N} x_i$ and $\alpha_2^{(N)} = \alpha_2 + N - \sum_{i=1}^{N} x_i$. Compared to the prior parameter values, they now contain information about the observations that we have made. We can use these new parameters to visualize our posterior for our initial set of coin flips and our initial choices of priors in Figure 2.1b. Similar to the frequentist confidence intervals of the previous sections, the uncertainty about the true value of θ is encoded in the spread of the posterior distribution. As we gather more observations, we expect the posterior to become more and more narrow around one (or few) values of θ. Similarly to the maximum likelihood estimate in Equation (2.15) that helps us determine the parameter value which is most likely to have generated our observations, we can derive a similar quantity in the Bayesian setting. This is referred to as the posteriori estimate (MAP), and can be interpreted as the most likely value of θ given the data and a choice of prior. We can derive the MAP using the posterior in Equation (2.21) and solving for θ:

$$
\begin{aligned}
\frac{\partial}{\partial \theta} \log p(\theta \mid \mathbb{D}) \overset{!}{=} 0 && (2.24)\\
(1 - \theta)\Big(\alpha_1 + \sum_{i=1}^{N} x_i - 1\Big) = \theta\Big(\alpha_2 + N - \sum_{i=1}^{N} x_i - 1\Big) && (2.25)\\
\hat{\theta}_{\text{MAP}} = \frac{\alpha_1 + \sum_{i=1}^{N} x_i - 1}{\alpha_1 + \alpha_2 + N - 2}. && (2.26)
\end{aligned}
$$

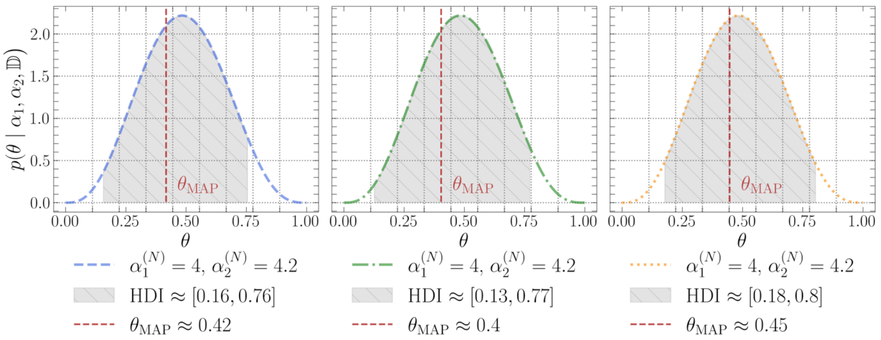


Figure 2.2: Highest density intervals (gray regions) and maximum a posteriori estimates (red vertical lines) for different Beta posteriors.

Highest Density Intervals. One way to now quantify the uncertainty about our estimate for θ is to create the Bayesian counterpart of confidence intervals: The *highest density interval* (HDI; also referred to as the *credible interval*). The HDI describes the ranges of values of the posterior distribution that covers 95% (or some other number) of the total density. Thus, our estimate has a posterior probability of 95% to fall within this interval. For the prior and posterior distributions in Figure 2.1, we obtain $\hat{\theta}_1 \approx 0.44$, $\text{HDI}_1 \approx [0.16, 0.76]$, $\hat{\theta}_2 \approx 0.43$, $\text{HDI}_2 \approx [0.13, 0.77]$, and $\hat{\theta}_3 \approx 0.49$, $\text{HDI}_3 \approx [0.18, 0.80]$, with the HDIs and MAP estimates shown in Figure 2.2. Since the second prior places less belief on a value of $\theta = 0.5$, the slightly skewed initial sample of coins $\hat{\theta} = 0.4$ shifts the posterior estimate and HDI slightly towards the left. In the third case, our prior belief is highly biased towards higher values of θ, which is also reflected in the obtained posterior estimate, the MAP estimate θ_{MAP} and its HDI. However, confidence intervals from Section 2.1.1 and the HDIs have very different interpretations, which echo the differences in frequentist and Bayesian thinking: The confidence intervals imply that, if we were to repeat our experiment 100 times, the true value for θ would be covered by the CIs 95 out of 100 times. In contrast, the HDI draws a conclusion about the range of values be believe the true parameter value to lie in, based on our prior belief updated using our actual observations.

Predictive Uncertainty. So far, we have discussed the uncertainty in our parameter estimate, but Bayesian statistics also provides a useful tool to reason about new observations: Predictive distributions. Let us assume we would like to make a prediction about the observation x' stemming from a new coin flip. We can write this probability as follows:

$$p(x') = \int_{\Theta} p(x' \mid \theta)p(\theta)\mathrm{d}\theta. \tag{2.27}$$

This is referred to as the *prior predictive distribution*, which gives us an instrument to reason about the outcome using the specified prior alone, disregarding any observations. One way to interpret this distribution is as a weighted aggregate of predictions for x' using different values of θ, which are weighed according to our prior belief. In the case of the Bernoulli and Beta distribution in the coin flip sample, this distribution has an analytical form:

$$\begin{aligned}
P(x') &= \int_{\Theta} P(x' \mid \theta)p(\theta)\mathrm{d}\theta && (2.28)\\
&= \int_0^1 \theta^{x'}(1-\theta)^{(1-x')}\frac{1}{\mathrm{B}(\alpha_1,\alpha_2)}\theta^{\alpha_1-1}(1-\theta)^{\alpha_2-1}\mathrm{d}\theta && (2.29)\\
&= \frac{1}{\mathrm{B}(\alpha_1,\alpha_2)}\int_0^1 \theta^{\alpha_1+x'-1}(1-\theta)^{(\alpha_2-x')}\mathrm{d}\theta && (2.30)\\
&= \frac{\mathrm{B}(\alpha_1+x',\alpha_2-x'+1)}{\mathrm{B}(\alpha_1,\alpha_2)}, && (2.31)
\end{aligned}$$

where the last step used the fact the Beta function can be expressed as $\mathrm{B}(\alpha_1,\alpha_2) = \int_0^1 x^{\alpha-1}(1-x)^{\alpha_2-1}\mathrm{d}x$ (see Appendix A.1 for more details). While it is useful to check whether a chosen prior is suitable for a given task, the prior predictive does not take any observations into account yet, so we would usually consider a predictive distribution given some available data. This is the purpose of the *posterior predictive distribution*, defined as

$$p(x' \mid \mathbb{D}) = \int_{\Theta} p(x' \mid \theta)p(\theta \mid \mathbb{D})\mathrm{d}\theta. \tag{2.32}$$

Again, we arrive at a prediction by "averaging" predictions made using different values of θ. Since not all values of θ are equally plausible given our data, they are furthermore weighted by their probability under the posterior $p(\theta \mid \mathbb{D})$. In a frequentist analysis, we only consider a single point estimation of $\hat{\theta}$ instead of a distribution. In terms of Equation (2.32), this can be expressed with the help of a Dirac delta function:

$$p(x' \mid \mathbb{D}) \approx \int_{\Theta} p(x' \mid \theta)\delta(\theta-\hat{\theta})\mathrm{d}\theta = p(x' \mid \hat{\theta}), \tag{2.33}$$

where we recover using only a single estimate $\hat{\theta}$ for our prediction. Back to the coin flip example, we can apply a similar argument as in Equation (2.28) with the posterior instead of the prior to arrive at

$$P(x' \mid \mathbb{D}) = \frac{\mathrm{B}(x'+\alpha_1^{(N)}, 1-x'+\alpha_2^{(N)})}{\mathrm{B}(\alpha_1^{(N)},\alpha_2^{(N)})}. \tag{2.34}$$

Interestingly, we can interpret the two terms in the right-hand side of Equation (2.32) as two different sources of uncertainty: The aforementioned uncertainty about the true value of θ given observed data is encoded in $p(\theta \mid \mathbb{D})$, and the uncertainty about x' given a fixed parameter value in $p(x' \mid \theta)$. This interpretation gives rise to the distinction of *data* (or *aleatoric*) uncertainty and *model* (or *epistemic*) uncertainty. The former usually refers to *irreducible* uncertainty that is inherent to the phenomenon we would like to model, like inherent ambiguity or unavoidable noise, and refers to $p(x' \mid \theta)$. The latter describes our uncertainty about the correct model parameters and resides in $p(\theta \mid \mathbb{D})$.[10] The more data we gather, the more we assume the posterior to be concentrated on only the most plausible parameter values, and thus the uncertainty is reduced. In the frequentist approach, tools like confidence intervals can only tell us about the total uncertainty of our estimate. In the Bayesian approach however, these different notions of uncertainty are represented by different distributions. These considerations are the basis for Bayesian deep learning methods, which we will discuss more in Section 2.2.2.

Recap. We have seen in this section how Bayesian statistics takes a very different approach to uncertainty than frequentist statistics: In frequentist statistics, probabilities are seen as relative frequencies of an event as we repeat an experiment. In Bayesian statistics, this interpretation is abandoned in favor of an viewpoint that sees probabilities as the degree of belief in an event, and parameters of interest becoming random variables instead of unobserved constants. It allows us to specify a prior belief which is updated using observations, and which diminishes in importance as we encounter more and more data. Furthermore, we can use predictive distributions to reason about unseen outcomes. In the posterior predictive distribution, we can also distinguish two kinds of uncertainty: Irreducible data uncertainty and model uncertainty, reducible by obtaining more data.

So far we have only discussed view of uncertainty from the perspective of statistics, defining models that explain observations and make new predictions. Despite their usefulness, it is no obvious how to apply these statistical models to phenomena as complex as human language, which we turn to next.

[10] Here, this categorization is approached from a general standpoint. We will discuss how these notions of uncertainty materialize in a language context in Section 2.3 and point out some problems and nuances.

2.1.3 The Linguistic Perspective: Underspecification, Ambiguity & Vagueness

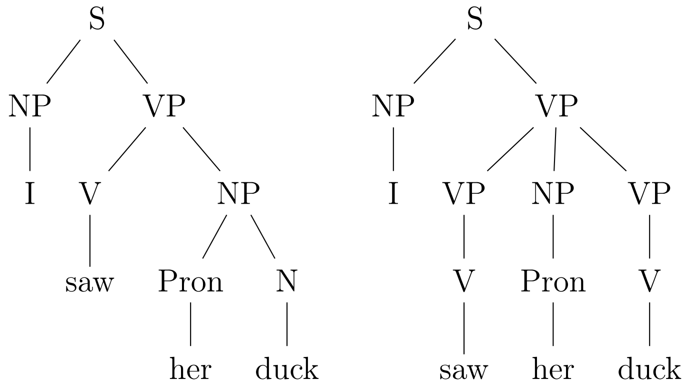
Linguistics can be categorized into multiple sub-disciplines that are concerned with different aspects of human language (Akmajian et al., 2017). This thesis focuses on written language, which is why we will not discuss any uncertainty in e.g. *phonetics* and *phonology* (the studies of the production of sounds and how they are organized in a language). Instead, we focus on the following three levels: *semantics*, *syntax* and *pragmatics*. In linguistics, uncertainty appears through different phenomena, for instance ambiguity or polysemy (Tuggy, 1993; Kennedy, 2011), underspecification (Pustejovsky, 1991, 2017) and vagueness (Tuggy, 1993; Brown, 2005; Kennedy, 2011), which manifests in different ways in different linguistic levels. This creates uncertainty by creating multiple different interpretations of a sentence, which are often—but not always—resolved through additional context, either linguistic, situational or from world knowledge. Describing this interplay between uncertainty and resolve on different linguistic levels is goal of this chapter.

Uncertainty in Semantics. The field of semantics is concerned with the literal meaning of words and the ways in which these are combined (Kearns, 2017). One way in which uncertainty arises in semantics is *polysemy*, a phenomenon where two or more distinct senses are associated with the same word (Gries, 2015). Gries for instance mentions the examples of "I emptied the glass" compared to "I drank a glass", where *glass* corresponds in the first case to a container, and to its content in the second. A more subtle case of polysemy is exemplified by the examples

(a) Jocelyn walked to the school.

(b) The concerned mother talked to the school.

where "school" in the former refers to the physical building, and the latter to the an administrative unit inside the organization that operates within the school building (Frisson, 2009). Resolving these cases can be highly non-trivial, leading in NLP to the field of *word sense disambiguation* (see e.g. Schütze, 1997; Agirre and Edmonds, 2007; Navigli, 2009). Another case is *homonymy*, where two unrelated meanings map onto the same form (Devos, 2003), as in the case of *bank* as a financial institute, a place for sitting, or the terrain alongside a river bed. *Vagueness* can be defined in contrast to these notions as whether "a piece of semantic information is part of the underlying semantic structure of the item, or the result of a contextual specification" or simply "the notion that certain features are not expressed in a representation" (Frisson, 2009; Geeraerts, 1993). In their example, they show how for "*my neighbor is a civil servant*", *neighbor* is not ambiguous since it does not require disambiguation in the given context, despite the word being underspecified (i.e., the neighbor's gender is for instance underspecified). Vagueness and underspecification are ubiquitous in language, since terms like *tall* or *red* are gradual and highly subjective terms (Brown, 2005) or simply because a speaker (or listener) is lacking information (Williamson, 2002). In addition, the meaning of some words might be underspecified unless or because it is combined with other words (Pustejovsky, 1991). The principle of *compositionality* states that the meaning of a more complex expression depends—completely or at least in part—on the meaning of its constituents (Fodor, 2001; Szabó, 2004; Brown, 2005). While composition of simpler to more complex expressions can help to resolve underspecification ("*my* ***female*** *neighbor is a civil servant*"), it can also create new underspecification, for instance through multiple quantifiers or prepositional phrases with multiple attachments (Pustejovsky, 2017, see next paragraph for an example).



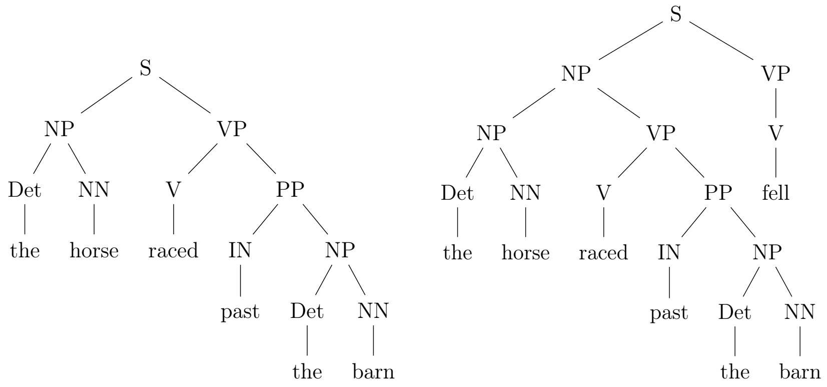
(a) Parsing *duck* as a noun. (b) Parsing *duck* as a verb.

Figure 2.3: Two equally valid parse trees for the sentence "I saw her duck" using a constituency grammar.

Uncertainty in Syntax. Syntax describes the machinery that combines the meaning of words and subwords into bigger units, such as phrases and sentences (Koeneman and Zeijlstra, 2017). In order to model this system, different grammatical formalisms have been proposed (Varile et al., 1997; section 3.3), which describe sets of rules that analyze a sentence in terms of a hierarchical structure that describes the relationship between words. These include *constituency grammars*, which will be used for illustrative purposes here. The core idea of this concept lies in observation that words can behave as either single units, or clump together to comprise units of meanings, called constituents (Jurafsky and Martin, 2022). Constituency grammars describe the rules according to which these constituents combine into more and more complex units of meanings. For instance, the phrase "the duck" consists of a *determiner* (Det), or article, "the", as well as a noun (N), "duck". Together, they are denoted as a *noun phrase*, or simply NP. In the same fashion, we can assign categories like pronoun (Pron), verb (V) and verb phrase (VP), that culminate in a sentence (S). An example of an analysis using a constituency grammar is given in Figure 2.3: Here, the words in the sentence "I saw her duck" are combined along these rules.[11] However, the word *duck* can be read both as the action of suddenly crouching and a word describing aquatic fowl. In this former interpretation, "her" is read as an object instead of a possessive pronoun. The corresponding parse tree is given in Figure 2.3a. The alternative reading as a possessive pronoun is shown in Figure 2.3b. By themselves, the two parse trees might be equally valid grammatical analyses given a constituency grammar. This implies that this *structural ambiguity* is unresolvable without any further context or world knowledge. Structural ambiguity can arise in a variety of situations depending on the language in question (see for instance Taha, 1983 for examples in English). Figure 2.3 depicts an attachment ambiguity: It is unclear whether *her* and *duck* attach as a combined NP to the VP of *saw*, or whether all three parts are equal constituents of a combined VP. Other popular examples include the attachment of (specifically) propositional phrases ("I saw the man with the telescope"; Schütze, 1995; Hindle and Rooth, 1990) or coordination ("old men and women"; Frazier et al., 2000; Engelhardt and Ferreira, 2010). Uncertainty can also appear in the processing of language when awaiting additional context. A famous example of this are *garden path sentences*, i.e. sentences that contain surprising syntactical elements that require a re-analysis of the sentence structure thus far (Sturt et al., 1999). The most famous example is the sentence "the horse raced past the barn fell", for which the corresponding syntactical parse trees are shown in Figure 2.4. Before observing the last word, the sentence in Figure 2.4a exhibits a simple structure of subject ("the horse"), verb ("raced") and a prepositional phrase ("past the barn"). After encountering "fell", we realize that "raced" was indeed not the main verb of the sentence, and instead is used to describe

[11] These rules can also defined more formally in the form of a *context-free grammar*, where rules are applied regardless of a context. The exact rules are omitted here for the sake of clarity, but it should be noted that is a simplifying assumption, as natural language is not context-free (Savitch et al., 2012).

that the horse that fell did so after having raced past the barn. Structurally, this requires the VP of "raced" in Figure 2.4b to be grouped under the subject NP, and a new VP to be created for "fell". Experimental evidence has shown that such ambiguous or challenging constructions can lead to an increase in human reading and processing times (see e.g. Milne, 1982; Ferreira and Henderson, 1991; Swets et al., 2008), suggesting that some form of re-analysis might occur.[12]



(a) Parse before the last word. (b) Parse after the last word.

Figure 2.4: Parse trees for the garden path sentence "The horse raced past the barn fell", before and after adding the last word, prompting a re-analysis of the sentence, where "raced past the barn" attached to the NP and "fell" becomes the new main verb.

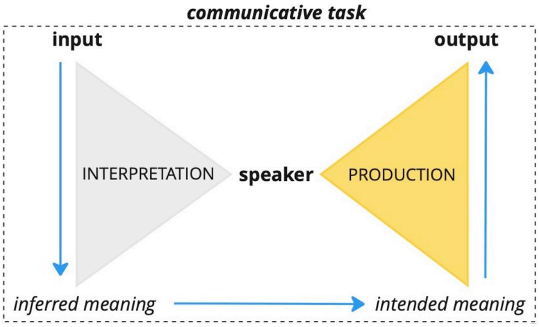


Figure 2.5: Double triangle of language production by Baan et al. (2023) as an extension of the triangle of reference by (Ogden and Richards, 1923).

Uncertainty in Pragmatics. Pragmatics can be defined as the study of language in use, especially in social interactions and speech

[12] Interestingly, similar effects have been observed in neural models (Van Schijndel and Linzen, 2018; Irwin et al., 2023), although the relationship is weaker in recent transformer models (Oh and Schuler, 2023; Oh et al., 2024).

(Mey, 2006; Huang, 2014). Compared to semantics, it also studies how word meanings are affected in the context of a specific utterance (Kearns, 2017). Baan et al. (2023) demonstrate its connection to uncertainty, specifically in natural language generation, through an extension of the "triangle of reference" by Ogden and Richards (1923), which is shown in Figure 2.5: Given an input to the speaker, there is a potentially wide set of possible inferred meanings; this can be caused by errors, underspecification (for instance where in some language the gender of a subject is not specified explicitly) or ambiguities of syntactical or semantic nature, as discussed in the previous paragraphs. This mapping from utterance to meaning is therefore not one-to-one, but rather one-to-many (Grice, 1957; Kennedy, 2011). As the speaker prepares their utterance, they then choose one of a variety of similar or even equivalent meaning to express the intended utterance. This production process is influenced by the speaker's social and cognitive idiosyncrasies (Levelt, 1993). We will refer to these two sources of uncertainty as *input* and *output variability* or *paraphrasticity* in the rest of thesis. This describes an important difference between language and other modalities: Since language is *paraphrastic*, there are (almost) equally valid ways to express the same intended meaning, which however might differ completely in their realizations, i.e. wordings.[13]

2.1.4 The Linguistic Perspective: Expressing Uncertainty

Besides paraphrastic language, a different type of uncertainty lies in explicit uncertainty expressions by the speaker. This spans the overall tone of a series of utterances to the usage of diverse linguistic expression (see e.g. Rubin, 2006 pp. 21–40; Lorson et al., 2023, Zhou et al., 2023), for instance *hedges* (Lakoff, 1973; Fraser, 1975; Prince et al., 1982; Holmes, 1982), i.e. words or phrases to express ambiguity or uncertainty. Additionally, uncertainty expressions might also be chosen circumstantially, for instance based on whether the other speaker is cooperative or uncooperative (Lorson et al., 2021), politeness (Sirota and Juanchich, 2015; Holtgraves and Perdew, 2016) or power differences between

[13] Some works argue against the concept that two expressions can be fully equivalent; for instance Widoff (2022) points out how two expressions can be equal in some form, but unequal in others (e.g. "the water in the glass" and "the glass is half-full" convey a similar meaning, but the second one does not specify the content) or how for instance instruments like passive voice can be be used to convey intent ("Hans beats Peter" vs. "Peter is beaten by Hans").

speakers (Bonnefon and Villejoubert, 2006).

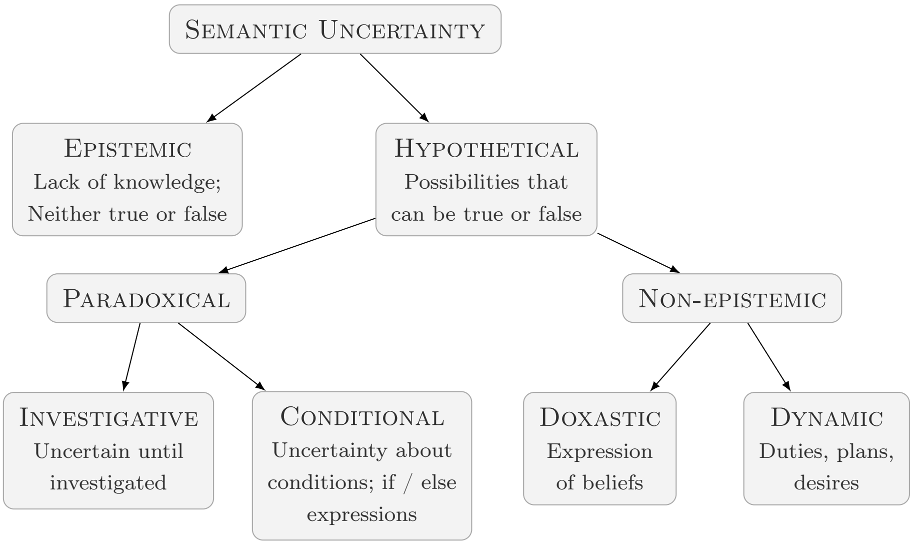


Figure 2.6: Taxonomy of different semantic uncertainties adapted from Kolagar and Zarcone (2024), originally based on the work by Szarvas et al. (2012).

Figure 2.6 shows a taxonomy of semantic uncertainties by Kolagar and Zarcone (2024), based on the works of Szarvas et al. (2012); Vincze (2014). It proposes a categorization of expressed uncertainty based on the truth value of an utterance. The taxonomy divides semantic uncertainty first into *epistemic*,[14] where the speaker expresses worlds which are neither true or untrue and do not coincide with their actual world. To make this notion less abstract, consider the following example: In the sentence "This is the best dessert I have ever had", we take the sentence to be a fact, and therefore assign a positive truth value. Now, we can instead use a modal verb to say "This may be the best dessert I have ever had". While one can imagine a possible world in which this statement is true, we cannot assign it a definitive truth value per se. The alternative branch in the taxonomy are *hypotheticals*, which can also be uncertain, but in contrast to epistemic uncertainty, also have the possibility of being evaluated as true or false. One bifurcation, *paradoxical*, refers to cases in which the truth value depends on another propositions, for instance if / else expressions (*conditional*) or cases in which the truth value can only be evaluated after further examination (*investigative*). The other fork, *non-epistemic*, describes circumstances in which a speaker expresses beliefs (*doxastic*) and duties, plans or desires

[14] Not be confused with the epistemic or model uncertainty in a statistical sense in Section 2.1.2.

(*dynamic*).

Recap. We have now discussed a variety of sources of uncertainty in linguistics, located on different levels of language use, including semantics, syntax and pragmatics. These discussions were mostly informed by phenomena in the English language and are thus limited, as other types of ambiguity exists that were not discussed here (see for instance Li et al., 2024a). We can nevertheless distill certain insights: On the one hand, uncertainty arises as an inherent property of language, through polysemy, structural ambiguities or possible paraphrases. On the other hand, uncertainty is a tool that be employed by a speaker to express their own uncertainty and to express the state of potential worlds. In both cases, this creates challenges for any processing system that operates on language and tries to infer its meaning.

2.1.5 A Pragmatic Answer

The astute reader might have noticed that while the title of Section 2.1 was "what is uncertainty, anyway?", it might appear that we have thus far been tiptoeing around this question, enumerating and explaining different perspectives to it without giving a satisfying answer.

In the end, uncertainty is a multifaceted and perhaps vague concept, whose definition varies based on the phenomenon of interest. At its very core, it describes a lack of knowledge about the true state of the world among competing alternative states. The definitions of these world states can differ tremendously on the context, and can include all the possible interpretations of the sentence "I saw her duck" to plausible values of a data-generating parameter θ. For the purpose of this thesis, we reduce its definition to the following aspects: Firstly, there is the uncertainty that is inherent to language described in Section 2.1.3, describing how interpretation and production are not a one-to-one processes of meaning; Secondly, the statistical models we apply to language are themselves faced with multiple possible specifications and can produce different potential predictions. As these models are at best informative but incomplete abstractions of reality that are fit on finite data, we accept their uncertainty as the price for practicality. While the last two points refer to uncertainty as phenomena, however and thirdly, uncertainty is also a tool: It enables us to reason about and express our own knowledge about possible states, and convey our lack thereof. This notion captures both the statistical sense, like considering different parameter values in the posterior predictive distribution, as well as making conditional statements or using linguistic modifiers to convey our beliefs in natural language. As NLP involves different kinds of uncertainty both in its modeling tools and data modality, this creates an intricate interplay between these uncertainties.

So far, we have looked at uncertainty in a fashion that is completely independent from neural networks, the core modeling tool of this thesis and main workhorse of contemporary artificial intelligence. Natural language processing specifically has adopted the use of large neural models operating on language inputs (and sometimes also outputs). It thus lies in the intersection of linguistics and statistics, and we will review the implications on uncertainty modeling next.

2.2 Uncertainty in Deep Learning

> "*In the 1950s and 60s, scientists built a few working perceptrons, as these artifical brains were called. [. . .] This perceptron is being trained to recognize the difference between males and females. [. . .] After training on lots of examples, it [. . .] is able to successfully distinguish male from female. It has learned. While promising, this approach to machine intelligence has virtually died out.*"
>
> —Clip about AI research in the 1950s and 60s, date unknown.

In contrast to the interviewer's quote in the epigraph, the very promising approach of using computational models of neurons did not completely die out, but rather remained dormant for decades. First known as *cybernetics* at the time of the first models of artificial neurons (McCulloch and Pitts, 1943; Rosenblatt, 1958; Rosenblatt et al., 1962), it became known as *connectionism* in the 1980–1990s, before assuming its current name *deep learning* in 2006 (Goodfellow et al., 2016). Nowadays, deep learning is commonly and vaguely defined as a family of machine learning networks that employ artificial neural networks of increasing depth (Goodfellow et al., 2016).[15] Uncertainty in deep learning materializes in a wide variety of approaches, as depicted as a hierarchical taxonomy in Figure 2.7: As shown in Section 2.2.1, the frequentist school uses neural networks

[15] One might argue that modern NLP might be at least in a large part subsumed by this definition; for this purpose of this thesis we will treat them as overlapping but different disciplines due to their history (see for instance Chapter 1 of Jurafsky and Martin, 2022) and the peculiarity of language as a data modality, especially compared to images or tabular data.

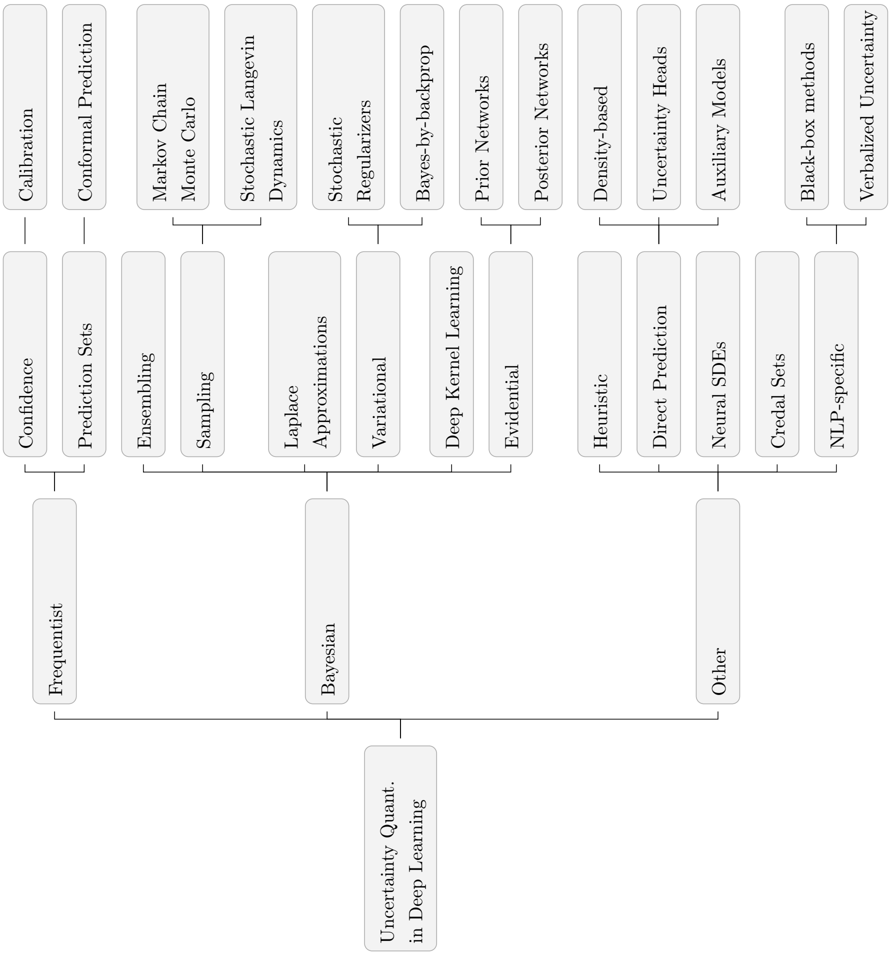


Figure 2.7: Hierarchical Taxonomy, showing the different methods discussed in Section 2.2. Note that these shown categories are not necessarily disjoint, as different methods can sometimes be placed into multiple categories at once.

as powerful estimators of predictive parameters, which can be interpreted similarly to the models in frequentist statistics we discussed earlier. In the same way, Bayesian methods can be applied to neural networks to quantify uncertainty through parameter posterior and posterior predictive distributions (Section 2.2.2). Other approaches take inspiration from the Dempster-Shafer theory of evidence (Section 2.2.3) or draw from entirely different ideas such as framing uncertainty quantification as a supervised learning task, stochastic differential equations, and more (Section 2.2.4).

2.2.1 Frequentist Neural Networks

Before we turn to how frequentist methods allow the quantification of uncertainty in neural networks, we first review the similarities in parameter estimation when applied to neural predictors. In the following, we term the application of frequentist methods to neural network as *frequentist neural networks*.

As introduced in Section 2.1.1, frequentist statistics refers to an interpretation of probability as the relative frequency of an event. In a neural network setting, the estimation of the parameter(s) of interest, in this case the network's parameters $\boldsymbol{\theta}$, is analogous to the maximum likelihood estimation in Section 2.1.1. The main differences are that firstly, instead of parameterizing a distribution with θ directly, we parameterize it with the prediction obtained from a neural net with parameters $\boldsymbol{\theta}$. Whereas in the coin flipping example, θ referred to the probability of heads, a neural network in a binary classification setting is equipped with some parameters $\boldsymbol{\theta}$ now predicts the probability of the positive class $\hat{p}$. And secondly, due to the model's non-linear and hierarchical dependencies, the solution to $\boldsymbol{\theta}$ is not available in closed form anymore. Instead, we iteratively optimize $\boldsymbol{\theta}$ through procedures such as gradient descent, where we compute the gradient of some loss function w.r.t. the parameters and take a step in the direction of the (anti-)gradient. The loss functions vary depending on the intended purpose, but in some cases can be directly related to maximum likelihood estimation. In analogy to the coin flipping in the previous section, a network trained on a binary classification task for instance predicts $\hat{p} = \sigma(f_{\boldsymbol{\theta}}(\mathbf{x}))$ (with $\sigma(\cdot)$ denoting the sigmoid function) and is then optimized using the binary cross-entropy loss (here for a single input using a gold label $y \in \{0, 1\}$):

$$\mathcal{L}_{\text{BCE}}(y, \hat{p}) = -y \log \hat{p} + (1 - y) \log(1 - \hat{p}). \tag{2.35}$$

The resemblance to the Bernoulli log-likelihood in Equation (2.9) is no coincidence, and we can see that the loss is minimized when the network prediction $\hat{p}$ correspond to the actual probability p, e.g. the relative occurrence of the positive class. Thus we view model predictions, at least for classification problems, through a similar, frequentist lens.[16]

Confidence & Calibration. To illustrate frequentist uncertainty estimation further, we now move from a binary classification problem to a multi-class classification problem. Formally, consider a neural predictor $f_{\boldsymbol{\theta}}$, a function mapping from an input space $\mathbb{R}^D$ to an output space $\mathbb{R}^K$ and with parameter vector $\boldsymbol{\theta}$. Here, K typically refers to the number of classes in a classification problem and the output of the network is referred to as *logits*. These logits are then normalized, typically by using the softmax function $\bar{\sigma}(\cdot)$, to produce a categorical probability distribution over classes. Since each class is now associated with a probability score, we can refer to each of these probabilities as the *confidence* of $f_{\boldsymbol{\theta}}$ regarding a certain class, or more formally

$$\hat{p}_k = P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x}) \equiv \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_k. \tag{2.36}$$

Also, let

$$\hat{y} = \operatorname*{argmax}_{k \in [K]} \hat{p}_k; \quad \hat{p} = \max_{k \in [K]} \hat{p}_k \tag{2.37}$$

be the class predicted by the model and its corresponding probability, respectively. Ideally, a predicted probability of e.g. 45% for some class would thus indicate that this is the correct prediction, in 45 out of 100 times, if we were to repeat the experiment. We can formulate this requirement as

$$p\big(y = \hat{y} \mid \hat{p}\big) = \hat{p}. \tag{2.38}$$

The degree to which this requirement is violated is measured through the expected calibration error (ECE; Naeini et al., 2015), which is defined as

$$\text{ECE} = \mathbb{E}\Big[\big|p\big(y = \hat{y} \mid \hat{p}\big) - \hat{p}\big|\Big]. \tag{2.39}$$

This expectation can for instance be computed by grouping N test predictions into M equally wide bins according to their confidence

[16] Here, we mainly focus on classification problems, which tend to be more frequent in NLP. Nevertheless, we can also consider a prediction from a trained regressor as frequentist by considering it as the mean of normal distribution with a variance equal to some (inverse and homoskedastic) noise, see for instance Bishop and Nasrabadi (2006), chapter 3.1.

$\hat{p}$. Defining $\mathcal{B}_m$ as the set indices that belong to bin m, we can write the ECE as

$$\text{ECE} \approx \sum_{m=1}^{M} \frac{|\mathcal{B}_m|}{N} \Bigg| \underbrace{\frac{1}{|\mathcal{B}_m|} \sum_{i \in \mathcal{B}_m} \mathbb{1}\big(\hat{y}_i = y_i\big)}_{\text{Bin accuracy (target)}} - \underbrace{\frac{1}{|\mathcal{B}_m|} \sum_{i \in \mathcal{B}_m} \hat{p}_i}_{\text{Avg. bin confidence}} \Bigg|, \tag{2.40}$$

where $\mathbb{1}\big(\hat{y}_i = y_i\big)$ is the indicator function showing whether the prediction was correct. As Guo et al. (2017) note, the terms in the difference approximate the left-hand and right-hand side in Equation (2.38) per bin, respectively. However, the ECE has also drawn several points of criticisms: The number of bins can also distort the results when test points are unequally distributed or an unsuitable number of bins is chosen. Also, it is not a *proper scoring rule* (Savage, 1971; Gneiting and Raftery, 2007), meaning that it is not necessarily minimized by the true distribution. For proper scoring rules, the true distribution should constitute the minimum, but for the ECE we can often minimize through a uniform distribution instead. Therefore, many other alternatives to the ECE have been proposed (e.g. Nixon et al., 2019; Kumar et al., 2019; Zhang et al., 2020a; Gruber and Buettner, 2022; Kirchenbauer et al., 2022; Roelofs et al., 2022; Błasiok and Nakkiran, 2023; Chidambaram et al., 2024).

Unfortunately, several works have shown that neural network models tend to be miscalibrated in general, with a tendency to be overconfident (e.g. Guo et al., 2017; Minderer et al., 2021; Zhu et al., 2023). Therefore, a vast library of methods has been proposed to improve the calibration of neural networks. This includes post-processing of predictions, for instance by retraining or adjusting the logits through additional scale and shift parameters (Platt et al., 1999; Guo et al., 2017; Mozafari et al., 2019; Kull et al., 2019; Wenger et al., 2020; Gupta et al., 2021; Ma and Blaschko, 2021). Others have introduced custom loss functions (Mukhoti et al., 2020b; Karandikar et al., 2021; Bohdal et al., 2021; Ghosh et al., 2022; Hebbalaguppe et al., 2022; Tao et al., 2023) that are meant to disincentivize overconfidence on a specific class. This is since performing maximum likelihood estimation of network parameters $\boldsymbol{\theta}$ with objectives such as Equation (2.35) only seeks to maximize the probability of the true class, but does not incentivize calibration per se. Other strategies involve tempering with the training data. Through label smoothing (Szegedy et al., 2016; Müller et al., 2019; Lukasik et al., 2020; Lienen and Hüllermeier, 2021b; Zhang et al., 2021a; Liu et al., 2022a; Park et al., 2023), where probability mass is dispersed from the ground truth class onto other classes, the network is taught to not assign maximal confidence to the ground truth. For further regularization, mixup can be used (Zhang et al., 2018b; Thulasidasan et al., 2019; Maroñas et al., 2021; Zhang et al., 2022a; Noh et al., 2023; Wang et al., 2023a), where the network is trained on interpolations of two inputs. In this case, both the input representations as well as their gold label distributions are mixed. Since miscalibration might also stem from a lack of training data, an intuitive way to improve models is data augmentation (Hendrycks et al., 2020; Patel et al., 2021). It has also been observed that ensembling (Lakshminarayanan et al., 2017; Wen et al., 2020a; Ashukha et al., 2020; Zhang et al., 2020a; Wu and Gales, 2021; Rahaman and Thiéry, 2021; Wen et al., 2021; Seligmann et al., 2024) and Bayesian modeling approaches (Mitros and Namee, 2019; Maroñas et al., 2020; Izmailov et al., 2021; Fortuin et al., 2022) can improve calibration (see Section 2.2.2).

Prediction Sets & Conformal Prediction. Instead of simply presenting a single prediction $\hat{y}$, we can also present the most likely outcomes in a *prediction set* instead, similar to a confidence interval. Let $\alpha \in [0, 1]$ be a hyperparameter controlling the desired width of a prediction set by defining a cutoff for probabilities. We can then define the prediction set $\mathcal{C}$ for a new point $\mathbf{x}'$ as

$$\mathcal{C}(\mathbf{x}') = \Big\{ y_{\pi^{-1}(1)}, \ldots, y_{\pi^{-1}(k')} \Big\} \tag{2.41}$$

$$k' = \sup \Big\{ k \ \Big| \ \sum_{j=1}^{k'} \hat{p}_{\pi^{-1}(j)} < 1 - \alpha \Big\} + 1. \tag{2.42}$$

The above formulation includes a sorting function $\pi(\cdot)$ that sorts indices k by their corresponding class probabilities $\hat{p}_k$, in a descending order, encompasses the most classes while staying under the probability threshold $1 - \alpha$, and adds one to avoid empty sets. Therefore, a more intuitive construction is the following: We sort all predicted probabilities from highest to lowest, and the select the classes for the prediction set until their sum exceeds a threshold of $1 - \alpha$. Ideally, we would like prediction sets to fulfil two criteria: They should contain the correct answer (*coverage*) and they should be as tight as possible.[17] $1-\alpha$ corresponds to the desired probability with which the correct answer should be contained in the prediction set in expectation, similarly how frequentist confidence scores correspond to a probability of correctness under many repetitions of an experiment. In this way, we can also interpret the width of the set as a proxy for confidence; The wider the set, the more uncertain

[17] Since one can always contain the correct answer by having the widest possible prediction sets, evaluating coverage alone is usually not meaningful.

the underlying model and the more classes it has to include in order to fulfill a coverage probability of $1-\alpha$. Unfortunately, and very similar to confidence scores, prediction sets are usually not calibrated by default (Kompa et al., 2021). The analogous solution to the calibration of prediction sets is *conformal prediction* (Vovk et al., 2005; Papadopoulos et al., 2002; Angelopoulos and Bates, 2021): By using a calibration set of data points and following the algorithm shown in Algorithm 1,[18] we can determine a probability threshold $\hat{q}$ in the following way: First, we collect a number of *non-conformity scores* s_i on a held-out calibration set that reflect the correctness of the model. The design of these scores is arbitrary, but should reflect the correctness of a model's prediction for a point, e.g. $s(\mathbf{x}_i) = 1 - p_{\boldsymbol{\theta}}(y_i \mid \mathbf{x}_i)$. Afterwards we choose $\hat{q}$ as the $\lceil (N+1)(1-\alpha)/N \rceil$-th quantile of the empirical score distribution. Using $\hat{q}$, our prediction sets now provably contain the correct prediction in expectation with a probability of $1-\alpha$. One simple way is to include all classes with a probability higher than $\hat{q}$:

$$\mathcal{C}(\mathbf{x}') = \big\{ y_k \;\big|\; \hat{p}_k \geq \hat{q} \big\}, \tag{2.43}$$

otherwise we can also repeat the construction in Equation (2.41), but replace the $1-\alpha$ threshold by $\hat{q}$. Prediction sets in this way then fulfil the following guarantee:

$$p\big(y' \in \mathcal{C}(\mathbf{x}')\big) \geq 1 - \alpha. \tag{2.44}$$

Algorithm 1 Conformal Prediction

```
Require: Calibration data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, predictor $p_{\boldsymbol{\theta}}$, non-
    conformity function $s : \mathbb{R}^D \to \mathbb{R}$ .

    ▷ 1. Retrieve non-conformity scores for calibration points, e.g.
    $s_i = s(\mathbf{x}_i) = 1 - p_{\boldsymbol{\theta}}(y_i \mid \mathbf{x}_i)$

    ▷ 2. Find quantile $\hat{q}$ using empirical inverse CDF $F_{\mathbb{S}}^{-1}$
    $\hat{q} \leftarrow F_{\mathbb{S}}^{-1}\big(\lceil (N+1)(1-\alpha)/N \rceil\big)$

    ▷ 3. Create prediction set, e.g.
    $\mathcal{C}(\mathbf{x}') \leftarrow \{y_k \mid \hat{p}_k \geq \hat{q}\}$
```

[18] This algorithm displays *split* conformal prediction, which can be applied to already trained predictors. *Full* conformal prediction however requires the re-training of the predictor on all the leave-one-out subsets of the training set, and is therefore infeasible for many modern settings. See for instance Angelopoulos and Bates (2021), section 6.

Conformal prediction has enjoyed great interest in recent years, since it is agnostic to the form of the underlying predictor and can therefore easily be applied to neural networks. Recent work has for instance be dedicated to apply conformal prediction for time series (Xu and Xie, 2021; Stankeviciute et al., 2021; Lin et al., 2022b; Zaffran et al., 2022) and other non-i.i.d. settings (Gibbs and Candès, 2021; Oliveira et al., 2022; Bhatnagar et al., 2023; Barber et al., 2023; Farinhas et al., 2024). It should also be noted that the conformal guarantee in Equation (5.2) can be rewritten in terms of the indicator function:

$$p\Big(\mathbb{1}\big(y' \in \mathcal{C}(\mathbf{x}')\big)\Big) \geq 1 - \alpha. \tag{2.45}$$

This fact is exploited by Angelopoulos et al. (2023) and subsequent works (Fisch et al., 2022; Farinhas et al., 2024; Xu et al., 2023b) to generalize this guarantee to families of functions that go beyond coverage, for instance controlling for false-negative rate (Angelopoulos et al., 2023; Fisch et al., 2022; Farinhas et al., 2024; Xu et al., 2023b).

Uncertainty Quantification in Frequentist Networks. In the case of prediction sets, their width can be interpreted as a confidence score: When the probability distribution is more uniform, more classes have to be added to the set to reach a specific probability threshold, and thus the set size grows. Without prediction sets, we turn to the (calibrated) confidence score, which is usually taken to be the maximum probability among all classes (Hendrycks and Gimpel, 2017). Alternatively, a popular measure of uncertainty is to compute the Shannon entropy of the distribution, which is given by

$$\mathrm{H}\big[P_{\boldsymbol{\theta}}(y \mid \mathbf{x})\big] = -\sum_{k=1}^{K} P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x}) \log P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x}). \tag{2.46}$$

The entropy is maximal when the distribution is uniform, and conversely its value is minimal when all the probability mass rests on a single outcome.

2.2.2 Bayesian Neural Networks

After reviewing the frequentist approach to neural networks in the previous section, the question naturally arises whether we can also apply Bayesian thinking in a neural network setting. This question can be answered affirmatively and has been studied since the 1990s (see e.g. Tishby and Solla, 1989; MacKay, 1992a; Neal, 1995).[19] We start by Bayesian parameter estimation for a neural network parameterized by weights $\boldsymbol{\theta}$. By placing a prior $p(\boldsymbol{\theta})$ over the weights, we obtain a posterior using Bayes' rule in Equation (2.16):

$$p(\boldsymbol{\theta} \mid \mathbb{D}) \propto p(\mathbb{D} \mid \boldsymbol{\theta})p(\boldsymbol{\theta}). \tag{2.47}$$

We can again find the maximum a posteriori estimate like in Section 2.1.2, but due to the nature of neural networks, have to resort to an iterative optimization procedure to find the parameters like for the neural maximum likelihood estimate in the previous Section 2.2.1. Luckily, we can optimize for $p(\boldsymbol{\theta} \mid \mathbb{D})$ by simply using a loss function such as in the previous section, and either explicitly or implicitly define a prior $p(\boldsymbol{\theta})$. Explicitly, this can be performed by for instance sampling the initial values of $\boldsymbol{\theta}$ from some prior distribution, or implicitly through regularization.[20] While that makes it comparatively easy to find the parameters $\boldsymbol{\theta}$ that maximize Equation (2.47), it is much harder to find the analytical form of the posterior $p(\boldsymbol{\theta} \mid \mathbb{D})$ or to sample from it. This is because the full form of Equation (2.47) derived via Bayes' rule includes the evidence term $p(\mathbb{D})$ as normalizing constant, which as shown in Equation (2.17), involves the marginalization over $\boldsymbol{\theta}$. This same infeasible marginalization also appears in the corresponding predictive distribution:

$$p(\mathbf{x}' \mid \mathbb{D}) = \int_{\Theta} p(\mathbf{x}' \mid \boldsymbol{\theta})p(\boldsymbol{\theta} \mid \mathbb{D})\mathrm{d}\boldsymbol{\theta}\,. \tag{2.48}$$

Why is this marginalization prohibitive? Compared to the conjugacy that allowed for the elegant solutions in Equations (2.15), (2.28) and (2.34), neural networks typically involve non-linear components in the form of activation functions, which enable their flexibility and modeling power as their depth increases (Hornik et al., 1989; Barron, 1994; Lu et al., 2017).[21] Formulating the likelihood $p(\mathbb{D} \mid \boldsymbol{\theta})$, this non-linear dependence of parameters makes it impossible to marginalize the parameters out. Numerical integration is also usually not feasible, since network parameters

[19] Due to the volume of the corresponding literature, we will restrict ourselves to some core ideas and important works, a brief history of the field can for instance be found in Gal (2016), pp. 20–23.

[20] Regularizing the l_2-norm of the network parameters for instance corresponds to the use of a isotropic normal prior (see e.g. Bishop and Nasrabadi, 2006; Section 3.3.1).

[21] As an illustrative counter-example, consider a simple two-layer network without non-linear activation functions in the form of

$$f(\mathbf{x}) = \begin{bmatrix} w_5 & w_6 \end{bmatrix} \left(\begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right),$$

are real-valued and typically high-dimensional. However, that does not mean that Bayesian inference with neural networks is impossible, it rather means that we have to employ a number of different strategies. A common red thread between them is that evaluating Equation (2.48) does not require us to have access to the distribution itself, only (high-quality) samples. As such, we can approximate the integral using Monte Carlo sampling:

$$p(\mathbf{x}' \mid \mathbb{D}) = \int_{\Theta} p(\mathbf{x}' \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathbb{D}) \mathrm{d}\boldsymbol{\theta} \approx \frac{1}{M} \sum_{m=1}^{M} p(\mathbf{x}' \mid \boldsymbol{\theta}^{(m)}), \quad (2.49)$$

where we assume access to a set $\{\boldsymbol{\theta}^{(m)}\}_{m=1}^{M}$ of M sampled parameter vectors. This Monte Carlo integration approximates Equation (2.48) with an error $1/\sqrt{M}$, that decreases as a function of the number of samples. It should be noted however that this approximation will be only asymptotically correct for samples from the true (and not an approximate) posterior, which we can obtain using the now following methods.

Markov Chain Monte Carlo & Stochastic Gradient Langevin Dynamics. In order to obtain representative samples from the posterior, we do not necessarily need the analytical form of the posterior. This idea is used by techniques such as *Markov chain Monte Carlo* (MCMC) and *stochastic gradient Langevin dynamics* (SGLD). In the case of MCMC, the core insight is that as long as we can evaluate $p(\boldsymbol{\theta} \mid \mathbb{D})$ up to the pesky evidence term $p(\mathbb{D})$, we can, in relative terms, determine whether one sample is more likely under the posterior than another. That means that upon formulating a suitable update rule, we can construct a chain of samples that leads from unlikely samples from $p(\boldsymbol{\theta} \mid \mathbb{D})$ to more likely ones. A thorough introduction to and overview over this family of methods is out of scope for this section, which is why we instead refer to (Robert et al., 1999) and the corresponding chapters in Bishop and Nasrabadi (2006); Gelman et al. (2021). The technique has found numerous applications for neural networks, e.g. Andrieu et al. (2003); Neal (1995); Cobb and Jalaian (2021); Li and Zhang (2023). Stochastic Gradient Langevin dynamics (SGLD) follows a similar intuition (Welling and Teh, 2011), however instead of formulating probabilistic transition rules, the constructed chain of samples follows the gradient of the prior and log-likelihood to seek posterior modes, similar to gradient descent. Trying to combine

which we can rewrite as $f(\mathbf{x}) = \mathbf{a}^{\mathrm{T}}\mathbf{x}$ with $a_1 = w_1 w_5 + w_3 w_6$ and $a_2 = w_2 w_5 + w_4 w_6$. Therefore, despite using two linear layers, we effectively obtain a single linear layer in practice, thus providing motivation for non-linear activation functions.

the advantages of both methods has even birthed SGLD / MCMC hybrids (Ma et al., 2015; Chen et al., 2016; Liu et al., 2016; Zhang et al., 2020b). In all cases, sampling methods remain challenging due to the high dimensional parameter space of neural networks and the often multi-modal nature of the weight posterior $p(\boldsymbol{\theta} \mid \mathbb{D})$.

Variational Inference. Instead of trying to sample from the posterior $p(\boldsymbol{\theta} \mid \mathbb{D})$, we can instead sample from an easier proposal distribution $q(\boldsymbol{\theta} \mid \boldsymbol{\phi})$ with parameters $\boldsymbol{\phi}$. For this proposal distribution to reasonably represent the original weight posterior, we try to minimize the difference between the two (Hinton and Van Camp, 1993; Graves, 2011). At first glance, this seems paradoxical—how can we minimize the distance from the posterior if we do not know its form? However, using the Kullback-Leibler (KL) divergence, we can rewrite this difference as follows:

$$
\begin{aligned}
&\min_{\boldsymbol{\phi}} \mathrm{KL}\big[q(\boldsymbol{\theta} \mid \boldsymbol{\phi}) \;\big|\big|\; p(\boldsymbol{\theta} \mid \mathbb{D})\big] && (2.50)\\
=&\min_{\boldsymbol{\phi}} - \int_{\Theta} q(\boldsymbol{\theta} \mid \boldsymbol{\phi}) \log \frac{q(\boldsymbol{\theta} \mid \boldsymbol{\phi})}{p(\boldsymbol{\theta} \mid \mathbb{D})} \mathrm{d}\boldsymbol{\theta} && (2.51)\\
=&\min_{\boldsymbol{\phi}} \int_{\Theta} q(\boldsymbol{\theta} \mid \boldsymbol{\phi}) \log \frac{q(\boldsymbol{\theta} \mid \boldsymbol{\phi})}{p(\mathbb{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})} \mathrm{d}\boldsymbol{\theta} && (2.52)\\
=&\min_{\boldsymbol{\phi}} \mathrm{KL}\big[q(\boldsymbol{\theta} \mid \boldsymbol{\phi}) \;\big|\big|\; p(\boldsymbol{\theta})\big] - \mathbb{E}_{q(\boldsymbol{\theta}\mid\boldsymbol{\phi})}\big[p(\mathbb{D} \mid \boldsymbol{\theta})\big]. && (2.53)
\end{aligned}
$$

To derive this expression, we exploited the fact that in Equation (2.52), the expectation of the evidence $p(\mathbb{D})$ under $q(\boldsymbol{\theta} \mid \boldsymbol{\phi})$ is a constant that does not influence the result of the optimization problem. Since this term is missing from the expression in Equation (2.53), we refer to the result as the *evidence lower bound* or ELBO. We can now evaluate the KL divergence in closed form when the proposal distribution and prior are chosen in a convenient form (e.g. Gaussian distributions), and the respective integrals can again be approximated via Monte Carlo approximation, and the parameters $\boldsymbol{\phi}$ be optimized via gradient descent. The only missing component is that sampling $\boldsymbol{\theta}$ from $q(\boldsymbol{\theta} \mid \boldsymbol{\phi})$ must be differentiable, which is achieved via the *reparameterization trick* (Opper and Archambeau, 2009; Kingma and Welling, 2014; Rezende et al., 2014). To show this, let $\boldsymbol{\phi} = \{\boldsymbol{\mu}, \boldsymbol{\rho}\}$ be the parameters of a Gaussian proposal distribution. Then we can obtain differentiable samples by

$$
\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}); \quad \boldsymbol{\theta} = \boldsymbol{\mu} + \boldsymbol{\rho} \circ \boldsymbol{\varepsilon}\,. \tag{2.54}
$$

After training, networks parameters can be sampled from $q(\boldsymbol{\theta} \mid \boldsymbol{\phi})$ directly to facilitate Bayesian neural networks and evaluate the predictive distribution in Equation (2.49). Examples for variational methods for Bayesian neural networks are given by Blundell et al. (2015); Hernández-Lobato and Adams (2015); Louizos and Welling (2016); Krueger et al. (2017); Pawlowski et al. (2017); Zhang et al. (2018a).

Stochastic Regularizers. Another line of research has been concerned with the interpretation of neural network regularizers as sources for stochastic network parameter samples. For instance, dropout (Srivastava et al., 2014) regularizes neural network weights by setting a random subset of them to zero. This is implemented by sampling a mask from a Bernoulli distribution with dropout probability p_{dropout} and multiplying it with the corresponding weight matrix $\mathbf{W} \in \mathbb{R}^{M \times N}$:[22]

$$\mathbf{W}_{\text{dropout}} = \mathbf{W} \circ \mathbf{M}; \quad \{\mathbf{M}_{ij}\}_{i,j=1}^{M,N} \sim \text{Bernoulli}(p_{\text{dropout}}). \tag{2.55}$$

As Gal and Ghahramani (2016b) argues, we can actually interpret a set of parameters $\boldsymbol{\theta}_{\text{dropout}}$ with dropout masks applied to it as a sample from a variational posterior; therefore, by using dropout at inference time (as opposed to just training time in its original form), we obtain a set of samples $\{\boldsymbol{\theta}^{(m)}_{\text{dropout}}\}_{m=1}^{M}$ that can be inserted back into the MC estimate of the predictive distribution in Equation (2.49). This technique is referred to as *Monte Carlo dropout* (or MC dropout) and has found a number of extensions over the years (Gal and Ghahramani, 2016a; Li and Gal, 2017; Gal et al., 2017a; Nalisnick et al., 2019a; Boluki et al., 2020; Durasov et al., 2021). A similar reasoning can be applied to batch normalization (Ioffe and Szegedy, 2015): Batch normalization works by normalizing the input $\mathbf{z}^{(l)}$ to a layer via an estimate of its mean and variance

$$\mathbf{z}^{(l)}_{\text{BN}} = \frac{\mathbf{z}^{(l)} - \mathbb{E}[\mathbf{z}^{(l)}]}{\sqrt{\text{Var}[\mathbf{z}^{(l)}] + \varepsilon}}, \tag{2.56}$$

where ε is a small value added to avoid numerical issues, and the mean and variance statistics are estimated empirically during training. Similar to MC dropout, Teye et al. (2018); Mukhoti et al. (2020a) re-interpret this as a source of stochasticity: By sampling a single batch from the training set at inference time, we can use it to set our batch statistics for expectation and variance. By using

[22] While the intuition of dropout lies in severing neural connections randomly, in practice it is often realized as an additional layer that is applied by zeroing out parts of activations. For instance, the parallel work of Blum et al. (2015) explores a variational objective using dropout that is applied directly to the activations.

this batch mean and variance for our current inference, Teye et al., we sample different hidden representations, than we interpreted as a result of the randomness in the underlying weights. In both cases, the advantages are obvious: These regularization components are already part of many deep learning architectures, and the only overhead added is by running multiple forward passes per test input, which—for smaller models—might only add negligible overhead. The more subtle downside lies in the fact variational inference techniques, as these techniques are counted as, tend to only explore limited regions of the posterior distribution (Wilson and Izmailov, 2020). As such, obtained samples might simply not be very representative of $p(\boldsymbol{\theta} \mid \mathbb{D})$ and lead to subpar predictions and uncertainty estimates.

Laplace Approximations. The idea of Laplace approximations can indeed by traced back to the eponymous Pierre-Simon Laplace (Laplace, 1774) and has been applied to deep learning first by MacKay (1992b). In order to approximate a complex distribution $p(\boldsymbol{\theta} \mid \mathbb{D})$, we first obtain a MAP estimate of the network parameters $\boldsymbol{\theta}_{\text{MAP}}$ as described in the beginning of this section. We then consider a second-order Taylor expansion for the loss function $\mathcal{L}(\mathbb{D}, \boldsymbol{\theta})$ at $\boldsymbol{\theta}_{\text{MAP}}$:

$$\begin{aligned}&\mathcal{L}(\mathbb{D}, \boldsymbol{\theta}) \approx \\ &\mathcal{L}(\mathbb{D}, \boldsymbol{\theta}_{\text{MAP}}) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}})^{\text{T}} \left(\nabla^2_{\boldsymbol{\theta}} \mathcal{L}(\mathbb{D}, \boldsymbol{\theta})\big|_{\boldsymbol{\theta}_{\text{MAP}}}\right)(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}}).\end{aligned} \tag{2.57}$$

By assuming that $\mathcal{L}(\mathcal{D}, \boldsymbol{\theta}_{\text{MAP}})$ is negligible for a fully trained network, we can identify

$$p(\boldsymbol{\theta} \mid \mathbb{D}) \approx \mathcal{N}\Big(\boldsymbol{\theta} \;\Big|\; \boldsymbol{\theta}_{\text{MAP}}, -\nabla^2_{\boldsymbol{\theta}} \mathcal{L}(\mathbb{D}, \boldsymbol{\theta})\big|_{\boldsymbol{\theta}_{\text{MAP}}}\Big)^{-1}\Big). \tag{2.58}$$

Unfortunately, the computation of the covariance matrix quickly becomes infeasible for larger models due to the quadratic nature of the Hessian. Therefore, different compromises have been proposed (Daxberger et al., 2021a), including last-layer approximations (Kristiadi et al., 2020; Snoek et al., 2015), approximation on subsets of weights (Daxberger et al., 2021b), factorizing the Hessian (Ritter et al., 2018a,b; Kristiadi et al., 2020; Yu et al., 2024; Bergamin et al., 2024), or variational approximations (Ortega et al., 2023). At inference time, network parameters can be drawn from the approximate posterior as with previous methods.

Ensembling. A long-existing method to boost predictive performance has been to train multiple predictors on a problem and to ensemble their outputs (Bauer and Kohavi, 1999; Dietterich, 2000). Combining predictions has already been studied since the late 1960s, e.g. in Bates and Granger (1969); Clemen (1989), with some works on neural network ensembles already in the 1990s (Hansen and Salamon, 1990; Levin et al., 1990; Liu and Yao, 1999; Zhou et al., 2002). After the deep learning revival, Lakshminarayanan et al. (2017) discovered that deep ensembles do not only improve generalization, but also tend to be well-calibrated and produce high-quality estimates of predictive uncertainty. While Lakshminarayanan et al. (2017) frame deep ensembles explicitly as non-Bayesian, Fort et al. (2019); Wilson and Izmailov (2020) later argued that ensembling actually *is* a form of Bayesian model averaging. Since the members of an ensemble are usually trained independently, they are better at converging to different solutions in the parameter space. Therefore, ensembles are argued to better represent the often multi-modal weight posterior than some of the methods discussed earlier like MC dropout or Laplace approximations, which rely on local approximations (Fort et al., 2019).

Naturally, the disadvantage of ensembling lies in having to train multiple predictors, which can be costly for modern, large neural neural networks. A flurry of research works has investigated alternatives to this costly procedure, such as having ensemble members share weights (Antorán et al., 2020; Liu et al., 2022b; Durasov et al., 2021; Laurent et al., 2023) or ensembling checkpoints of a model collected over the training (Izmailov et al., 2018; Maddox et al., 2019; Izmailov et al., 2019; Wilson and Izmailov, 2020; Yashima et al., 2022). As our understanding of neural loss landscapes improves, works such as Garipov et al. (2018); Cha et al. (2021) have suggested to create ensembles along low-loss basins. Other ways to curb computational inference costs involve efficient weight factorization techniques (Wenzel et al., 2020; Wen et al., 2020b; Dusenberry et al., 2020) or distilling properties of an entire ensemble into a single predictor (Malinin et al., 2020; Kim et al., 2024a). Even when ensemble members are trained independently, they can converge to similar solutions, offsetting their advantage. Several methods to improve the diversity in ensembles have been proposed (Jain et al., 2020; D'Angelo and Fortuin, 2021; El-Laham et al., 2023), including the ensembling of different architectures (Zaidi et al., 2021). Notable are also other explicitly Bayesian ways of ensembling (Pearce et al., 2020; Deng et al., 2022) or connections to mixture-of-experts models (Allingham et al., 2022).

Deep Kernel Learning. Gaussian processes (GP; Kolmogoroff, 1941; Wiener, 1949; Williams and Rasmussen, 2006) are (typically) non-parametric models that predict targets and corresponding uncertainties based on the similarity between training and test points. These similarities are computed through covariance or kernel functions. In theory, this creates appealing properties for uncertainty quantification, as unusual inputs should be labeled as uncertain because of their dissimilarity with the observed data. Nevertheless, scaling Gaussian processes to large amounts of data in known to be challenging (see e.g. discussions in Williams and Rasmussen, 2006 or in Bishop and Nasrabadi, 2006, Chapter 6). Therefore, *deep kernel learning* (Wilson et al., 2016) fits a Gaussian process layer on top of a deep neural feature extractor. This has created a number of follow-up works using deep kernel learning for UQ (Bradshaw et al., 2017; Daskalakis et al., 2020; Liu et al., 2021; van Amersfoort et al., 2021), however several authors have noted shortcomings with the approach due to the challenging joint optimization of the GP and neural feature extractor (Ober et al., 2021; van Amersfoort et al., 2021; Schwöbel et al., 2022): This includes overfitting and feature collapse, where OOD data points are mapped to similar regions of the latent space as training points. On top of deep kernel learning, there are several connections between neural networks and GPs are given through deep Gaussian processes (Damianou and Lawrence, 2013; Dunlop et al., 2018; Jakkala, 2021) and the theoretical links between neural networks and GPs (Neal, 1995; Williams, 1998; Hensman and Lawrence, 2014; Dutordoir et al., 2021).

Uncertainty Quantification in Bayesian Networks. So far, we have discussed multiple different methods how to obtain samples from the (approximate) weight posterior, but without mentioning how this aids in obtaining new, useful and disentangled uncertainties. One way to assess epistemic uncertainty in this framework is to measure disagreement between predictions for the same input. Since models tend to be underspecified on OOD inputs, this is where the predictions from different models will disagree the most in case of high model uncertainty. In classification, this can be done for instance using the *variation ratio* (Freeman, 1965; Gal, 2016): Assuming a set of B samples from the weight posterior, let $\hat{y}^{(b)}$ the predicted label using each set of weights and let y^* denote the most commonly predicted label among these. Then, the variation-ratio is defined as

$$\mathrm{VR} = 1 - \frac{1}{B}\mathbb{1}\big(\hat{y}^{(b)} = y^*\big). \tag{2.59}$$

Another way to measure the disagreement between predictions is to simply quantify the average variance of predictions per class:

$$\bar{\sigma}^2 = \frac{1}{K}\sum_{K=1}^{K} \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\big[P_{\boldsymbol{\theta}}(y=k \mid \mathbf{x})^2\big] - \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\big[P_{\boldsymbol{\theta}}(y=k \mid \mathbf{x})\big]^2. \tag{2.60}$$

A more theoretically motivated approach to isolate epistemic uncertainty is to consider the mutual information between model parameters and a data sample (Depeweg et al., 2018; Smith and Gal, 2018):

$$\underbrace{\mathrm{I}\big[y, \boldsymbol{\theta} \mid \mathbb{D}, \mathbf{x}\big]}_{\text{Model uncertainty}} = \underbrace{\mathrm{H}\Big[\mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\big[P_{\boldsymbol{\theta}}(y \mid \mathbf{x})\big]\Big]}_{\text{Total uncertainty}} - \underbrace{\mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\Big[\mathrm{H}\big[P_{\boldsymbol{\theta}}(y \mid \mathbf{x})\big]\Big]}_{\text{Data uncertainty}}. \tag{2.61}$$

The term itself can be interpreted as the gain in information about the ideal model parameters and correct label upon receiving an input. If we can only gain a little, that implies that parameters are already well-specified and that the epistemic uncertainty is low. In both cases of Equations (4.3) and (4.4) the expectation can be approximated through Monte Carlo approximation, i.e.

$$\mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\big[P_{\boldsymbol{\theta}}(y=k \mid \mathbf{x})\big] \approx \frac{1}{B}\sum_{b=1}^{B} P(y=k \mid \mathbf{x}, \boldsymbol{\theta}^{(b)}). \tag{2.62}$$

2.2.3 Evidential Neural Networks

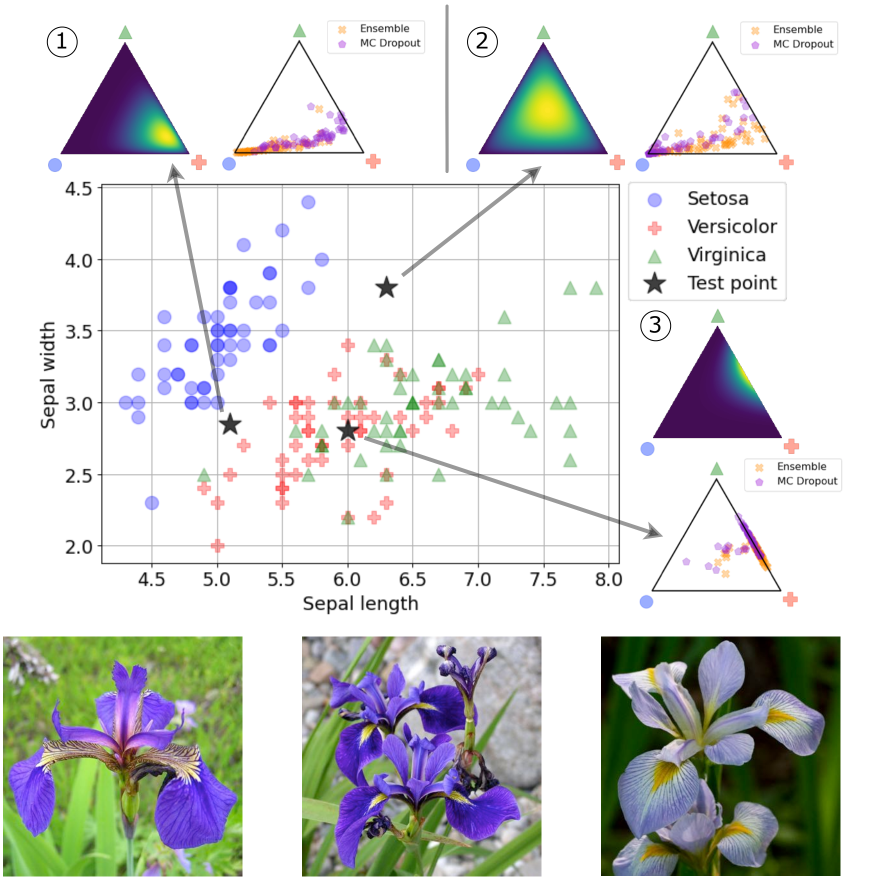
The following work is based on Ulmer et al. (2023).

In the last section, we explored many different approaches to quantify different kinds of uncertainty by obtaining samples from the weight posterior $p(\boldsymbol{\theta} \mid \mathbb{D})$. However, we saw that this can be a challenging endeavor, since samples might be expensive to obtain or not very representative of the actual posterior distribution. Alternatively, we can factorize Equation (2.48) further and use a point estimate for the weights to obtain a tractable form:

$$
\begin{aligned}
P(y \mid \mathbf{x}, \mathbb{D}) &= \int p(\mathbf{x}' \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathbb{D}) \mathrm{d}\boldsymbol{\theta} && (2.63)\\
&= \iint \underbrace{P(y \mid \boldsymbol{\pi})}_{\text{Aleatoric}} \underbrace{p(\boldsymbol{\pi} \mid \mathbf{x}, \boldsymbol{\theta})}_{\text{Distributional}} \underbrace{p(\boldsymbol{\theta} \mid \mathbb{D})}_{\text{Epistemic}} \mathrm{d}\boldsymbol{\pi} \mathrm{d}\boldsymbol{\theta} && (2.64)\\
&\approx \int P(y \mid \boldsymbol{\pi}) \underbrace{p(\boldsymbol{\pi} \mid \mathbf{x}, \hat{\boldsymbol{\theta}})}_{p(\boldsymbol{\theta}|\mathbb{D}) \approx \delta(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})} \mathrm{d}\boldsymbol{\pi}\,. && (2.65)
\end{aligned}
$$

In the last step, Malinin and Gales replace $p(\boldsymbol{\theta} \mid \mathbb{D})$ by a point estimate $\hat{\boldsymbol{\theta}}$ using the Dirac delta function, i.e. a single trained neural network, to get rid of the intractable integral.[23] This factorization contains another type of uncertainty, which Malinin and Gales (2018) call the *distributional* uncertainty; uncertainty caused by the mismatch of training and test data distributions. Although another integral remains, retrieving the uncertainty from this predictive distribution actually has a closed-form analytical solution, as we will see later. The advantage of this approach is further that it allows us to distinguish uncertainty about a data point because it is ambiguous, from uncertainty caused by a point coming from an entirely different data distribution. This approach to UQ it called *evidential deep learning* (EDL), and originates from the work of Sensoy et al. (2018). They originally base their motivation on the *theory of evidence* (Dempster, 1968; Audun, 2018): Within the theory, belief mass is assigned to set of possible states, e.g. class labels, and can also express a lack of evidence, i.e. an "I don't know". We can apply this idea to the predicted output of a neural classifier using the Dirichlet distribution, allowing us to express a lack of evidence through a uniform Dirichlet. In this way, the neural network does not parameterize a single (categorical) distribution, but a *distribution over distributions*, also referred to as a second-order distribution. This is different from a uniform (first-order) categorical distribution, which does not distinguish an equal probability for all classes from a lack of evidence, or differently phrased: One cannot distinguish whether the distribution is uniform due to uncertainty, or confidently uniform due to ambiguity. In the following, we define EDL as a family of approaches in which a neural network can fall back onto a uniform prior for unknown inputs. While neural networks usually parameterize like-

[23] In the context of Equation (2.63), it should be noted that restricting oneself to a point estimate of the network parameters prevents the epistemic uncertainty estimation through the weight posterior $p(\boldsymbol{\theta} \mid \mathbb{D})$, as discussed in the previous section. However, there are works like Haussmann et al. (2019); Zhao et al. (2020) that combine both approaches.



(a) *Iris setosa* (b) *Iris versicolor* (c) *Iris virginica*

Figure 2.8: Illustration of different approaches to uncertainty quantification on the Iris dataset, with examples for the classes given on the left (Figures 2.8a to 2.8c). On the right, the data is plotted alongside some predictions of a prior network (lighter colors indicate higher density) and an ensemble and MC dropout model on the probability simplex, with 50 predictions each. Iris images were taken from Wikimedia Commons, 2022a,b,c.

lihood functions, approaches in this survey parameterize prior or posterior distributions instead, as we will show next.

An Illustrating Example: The Iris Dataset. We train a deep neural network ensemble (Lakshminarayanan et al., 2017) with 50 model instances, a model with MC Dropout (Gal and Ghahramani, 2016b) with 50 predictions and a prior network (Sensoy et al., 2018), an example of EDL, on all available data points, and plot their predictions on three test points on the probability simplex in Figure 2.8.[24] On these simplices, each point signifies a categorical distribution, with the proximity to one of the corners indicating

[24] For information about training and model details, see Appendix C.4.1.

a higher probability for the corresponding class. EDL methods for classification do not predict a single output distribution, but an entire *density over output distributions*. Test point ③ lies in a region of overlap between instances of *Iris versicolor* and *Iris virginica*, thus inducing high aleatoric uncertainty. In this case, we can see that the prior network places all of its density around the vertex between these two classes, similar to most of the predictions of the ensemble and MC dropout (bottom right). However, some of the latter predictions still land in the center of the simplex. The point ① is located in an area without training examples between instances of *Iris versicolor* and *setosa*, as well as close to a single *virginica* outlier. As shown in the top left, ensemble and MC dropout predictions agree that the point belongs to either the *setosa* or *versicolor* class, with a slight preference for the former. The prior network concentrates its prediction on *versicolor*, but admits some uncertainty towards the two other choices. The last test point ② is placed in an area of the feature space devoid of any data, roughly equidistant from the three clusters of flowers. Similar to the previous example, the ensemble and MC dropout predictions on the top right show a preference for *Iris setosa* and *versicolor*, albeit with higher uncertainty. The prior network however shows an almost uniform density, admitting distributional uncertainty about this particular input. This simple example provides some insights into the potential advantages of EDL: First of all, the prior network was able to provide reasonable uncertainty estimates in comparison with Bayesian model averaging methods. Secondly, the prior network is able to admit its lack of knowledge for the OOD data point by predicting an almost uniform prior, something that the other models are not able to. Lastly, training the prior network only required a single model, which is a noticeable speed-up compared to MC dropout and especially the training of ensembles.

Parameterization. We start from a categorical distribution over classes, defined as:

$$\text{Categorical}(y \mid \boldsymbol{\pi}) = \prod_{k=1}^{K} \pi_k^{\mathbb{1}\left(y=k\right)}, \tag{2.66}$$

in which K denotes the number of categories or classes, and the class probabilities are expressed using a vector $\boldsymbol{\pi} \in [0,1]^K$ with $\sum_k \pi_k = 1$, and $\mathbb{1}\left(\cdot\right)$ is the indicator function. In this setting, the Dirichlet distribution arises as a suitable prior and multivariate generalization of the Beta distribution (and is thus also called the *multivariate Beta distribution*):

$$\mathrm{Dir}(\boldsymbol{\pi};\boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})}\prod_{k=1}^{K}\pi_k^{\alpha_k-1}; \quad \mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{k=1}^{K}\Gamma(\alpha_k)}{\Gamma(\alpha_0)}; \quad \alpha_0 = \sum_{k=1}^{K}\alpha_k, \tag{2.67}$$

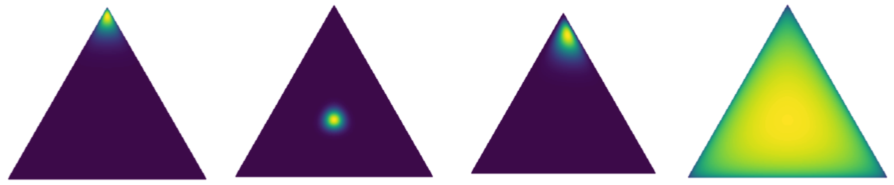
where $\alpha_k \in \mathbb{R}^+$ and the Beta function $\mathrm{B}(\cdot)$ is defined for K shape parameters compared to Equation (2.18). The distribution is characterized by its *concentration parameters* $\boldsymbol{\alpha}$, the sum of which, often denoted as α_0, is called the *precision*.[25] The Dirichlet is a *conjugate prior* for such a categorical likelihood, meaning that according to Bayes' rule, it produces a Dirichlet posterior with parameters $\boldsymbol{\beta}$, given a data set $\mathbb{D} = \{(x_i, y_i)\}_{i=1}^{N}$ of N observations with corresponding labels:

$$\begin{aligned}
p(\boldsymbol{\pi} \mid \mathbb{D}, \boldsymbol{\alpha}) &\propto p\big(\{y_i\}_{i=1}^{N} \mid \boldsymbol{\pi}, \{x_i\}_{i=1}^{N}\big)p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}) \\
&= \prod_{i=1}^{N}\prod_{k=1}^{K}\pi_k^{\mathbb{1}\left(y_i=k\right)}\frac{1}{\mathrm{B}(\boldsymbol{\alpha})}\prod_{k=1}^{K}\pi_k^{\alpha_k-1} && (2.68) \\
&= \prod_{k=1}^{K}\pi_k^{\left(\sum_{i=1}^{N}\mathbb{1}\left(y_i=k\right)\right)}\frac{1}{\mathrm{B}(\boldsymbol{\alpha})}\prod_{k=1}^{K}\pi_k^{\alpha_k-1} && (2.69) \\
&= \frac{1}{\mathrm{B}(\boldsymbol{\alpha})}\prod_{k=1}^{K}\pi_k^{N_k+\alpha_k-1} \propto \mathrm{Dir}(\boldsymbol{\pi};\boldsymbol{\beta}), && (2.70)
\end{aligned}$$

where $\boldsymbol{\beta}$ is a vector with $\beta_k = \alpha_k + N_k$, with N_k denoting the number of observations for class k. Intuitively, this implies that the prior belief encoded by the initial Dirichlet is updated using the actual data, sharpening the distribution for classes for which many instances have been observed. The Dirichlet is a *distribution over categorical distributions* on the $K-1$ probability simplex—while a neural classifier is usually realized as a function $f_{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}^K$, mapping an input $\mathbf{x} \in \mathbb{R}^D$ to *logits* for each class. Followed by a softmax function, this then defines a categorical distribution over classes with a vector $\boldsymbol{\pi}$ with $\pi_k \equiv P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})$. The same underlying architecture can be used without any major modification to instead parameterize a Dirichlet, predicting a distribution over categorical distributions $p(\boldsymbol{\pi} \mid \mathbf{x}, \hat{\boldsymbol{\theta}})$ as in Equation (2.67).[26] In order to classify a data point $\mathbf{x}$, a categorical distribution is created from

[25] The precision is analogous to the precision of a Gaussian, where a larger α_0 signifies a sharper distribution.

[26] The only thing to note here is that the every α_k has to be strictly positive, which can for instance be enforced by using an additional softplus, exponential or ReLU function (Sensoy et al., 2018; Malinin and Gales, 2018; Sensoy et al., 2020).



(a) Confident prediction. (b) Aleatoric uncertainty. (c) Epistemic uncertainty. (d) Distributional uncertainty.

Figure 2.9: Examples of the probability simplex for a $K = 3$ classification problem, where every corner corresponds to a class and every point to a categorical distribution, and brighter colors correspond to higher density. Shown is the (desired) Behavior of Dirichlet in different scenarios by Malinin and Gales (2018): (a) For a confident prediction, the density is concentrated in the corner of the simplex corresponding to the assumed class. (b) In the case of aleatoric uncertainty, the density is concentrated in the center, and thus uniform categorical distributions are most likely. (c) In the case of model uncertainty, the density may still be concentrated in a corner, but more spread out, expressing the uncertainty about the right prediction. (d) In the case of an OOD input, a uniform Dirichlet expresses that any categorical distribution is equally likely, since there is no evidence for any known class.

the predicted concentration parameters of the Dirichlet as follows (this corresponds to the mean of the Dirichlet, see Appendix A.2):

$$\boldsymbol{\alpha} = \exp\big(f_{\boldsymbol{\theta}}(\mathbf{x})\big); \quad \pi_k = \frac{\alpha_k}{\alpha_0}; \quad \hat{y} = \underset{k \in [K]}{\operatorname{argmax}}\ \pi_1, \ldots, \pi_K. \tag{2.71}$$

Uncertainty Quantification in EDL. Let us now turn our attention to how to estimate the aleatoric, epistemic and distributional uncertainty within the Dirichlet framework. In Figure 2.9, we show different (ideal) shapes of a Dirichlet distribution parameterized by a neural network, corresponding to different cases of uncertainty, where each point on the simplex represents a categorical distribution, with proximity to a corner indicating a high probability for the corresponding class. However, since we do not want to inspect Dirichlets visually, we instead use closed-form expressions to quantify uncertainty. To obtain a measure of data uncertainty, we can evaluate the expected entropy of the data distribution $P(y \mid \boldsymbol{\pi})$. As the entropy captures the "peakiness" of the output distribution, a lower entropy indicates that the model is concentrating most probability mass on a single class, while high entropy characterizes a more uniform distribution—the model is undecided about the right prediction. For Dirichlet networks, this quantity has a closed-form solution (for the full derivation, refer to Appendix A.4):

$$\mathbb{E}_{p(\boldsymbol{\pi}|\mathbf{x},\hat{\boldsymbol{\theta}})}\Big[\mathrm{H}\big[P(y \mid \boldsymbol{\pi})\big]\Big] = -\sum_{k=1}^{K}\frac{\alpha_k}{\alpha_0}\Big(\psi(\alpha_k{+}1){-}\psi(\alpha_0{+}1)\Big), \quad (2.72)$$

where ψ denotes the digamma function, defined as $\psi(x) = \frac{d}{dx}\log\Gamma(x)$, and H the Shannon entropy. As we saw in Equation (2.63), we can avoid the intractable integral over network parameters $\boldsymbol{\theta}$ by using a point estimate $\hat{\boldsymbol{\theta}}$.[27] This means that computing the model uncertainty via the weight posterior $p(\boldsymbol{\theta} \mid \mathbb{D})$ like in Section 2.2.2 is not possible. Nevertheless, a key property of Dirichlet networks is that epistemic uncertainty is expressed through the spread of the Dirichlet distribution (for instance in Figure 2.9 (c) and (d)). Therefore, the epistemic uncertainty can be quantified considering the concentration parameters $\boldsymbol{\alpha}$ that shape this distribution: Charpentier et al. (2020) simply consider the maximum α_k as a score akin to the maximum probability score by Hendrycks and Gimpel (2017), while Sensoy et al. (2018) compute it by $K/\sum_{k=1}^{K}(\alpha_k + 1)$ or simply α_0 (Charpentier et al., 2020). In both cases, the underlying intuition is that larger α_k produce a sharper density, and thus indicate increased confidence in a prediction. Lastly, the distributional uncertainty can be quantified by computing the difference between the total amount of uncertainty and the data uncertainty (similar to the reasoning behind Equation (4.4)), which can be expressed through the mutual information between the label y and its categorical distribution $\boldsymbol{\pi}$:

$$\mathrm{I}\big[y, \boldsymbol{\pi} \mid \mathbf{x}, \mathbb{D}\big] = \underbrace{\mathrm{H}\Big[\mathbb{E}_{p(\boldsymbol{\pi}|\mathbf{x},\mathbb{D})}\big[P(y \mid \boldsymbol{\pi})\big]\Big]}_{\text{Total Uncertainty}} - \underbrace{\mathbb{E}_{p(\boldsymbol{\pi}|\mathbf{x},\mathbb{D})}\Big[\mathrm{H}\big[P(y \mid \boldsymbol{\pi})\big]\Big]}_{\text{Data Uncertainty}}. \quad (2.73)$$

This quantity expresses how much information we would receive about $\boldsymbol{\pi}$ if we were given the label y, conditioned on the new input $\mathbf{x}$ and the training data $\mathbb{D}$. In regions in which the model is well-defined, receiving y should not provide much new information about $\boldsymbol{\pi}$—and thus the mutual information would be low. Yet, such knowledge should be very informative in regions in which few data have been observed, and there this mutual information would indicate higher distributional uncertainty. Given that $\mathbb{E}[\pi_k] = \frac{\alpha_k}{\alpha_0}$ (Appendix A.2) and assuming the point estimate $p(\boldsymbol{\pi} \mid \mathbf{x}, \mathbb{D}) \approx p(\boldsymbol{\pi} \mid \mathbf{x}, \hat{\boldsymbol{\theta}})$ to be sufficient (Malinin and Gales, 2018), we obtain an expression very similar to Equation (2.72):

[27] When the distribution over parameters in Equation (2.63) is retained, alternate expressions of the aleatoric and epistemic uncertainty are derived by Woo (2022).

$$\mathrm{I}\big[y, \boldsymbol{\pi} \mid \mathbf{x}, \mathbb{D}\big] = -\sum_{k=1}^{K} \frac{\alpha_k}{\alpha_0}\Big(\log \frac{\alpha_k}{\alpha_0} - \psi(\alpha_k+1) + \psi(\alpha_0+1)\Big). \quad (2.74)$$

We mentioned before how Figure 2.9 illustrates idealized behaviors of the Dirichlet distributions. Therefore, any closed-form expressions of different uncertainties can only be effective when the desired shape of the distribution is attained. Similarly, the naive parameterization in Equation (2.71) is not to guaranteed to succeed in this goal, and the literature has proposed different methods to attain this goal. They can broadly be classified into two families: *Prior networks*, which parameterize the Dirichlet prior distribution and employ custom training procedures and regularizers, and *posterior networks*, which instead parameterize a Dirichlet posterior like in Equation (2.68) instead.[28]

Prior Networks. Prior networks can be further subcategorized into two sets, namely OOD-free approaches or OOD-dependent approaches. In the first case, we regulate the behavior of the Dirichlet distribution on OOD inputs by adding a regularizer that penalizes any density allocated to regions that do not correspond to the gold label. One such option is to decrease the Kullback-Leibler divergence from a uniform Dirichlet (see Appendix A.5):

$$\mathrm{KL}\big[p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}) \,\big|\big|\, p(\boldsymbol{\pi} \mid \mathbf{1})\big] = \log \frac{\Gamma(K)}{\mathrm{B}(\boldsymbol{\alpha})} + \sum_{k=1}^{K} (\alpha_k - 1)\big(\psi(\alpha_k) - \psi(\alpha_0)\big). \quad (2.75)$$

Other options are the use of Rényi divergences (Tsiligkaridis, 2019), regularizers derived from PAC-bounds (Haussmann et al., 2019), or l_p-norms (Sensoy et al., 2018; Tsiligkaridis, 2019). Alternatively, some works also try to transfer the uncertainty from a set of Bayesian predictors into a single prior network (Malinin et al., 2020; Fathullah and Gales, 2022) using knowledge distillation (Hinton et al., 2015). When OOD data is available, we also explicitly train the prior network to maximize its entropy on such examples (Malinin and Gales, 2018, 2019; Nandy et al., 2020), which can for instance be implemented using the closed-form solution in Appendix A.3:

[28] We now give a brief overview over these approaches with a focus on classification problems. For a more comprehensive account that also includes regression problems, refer to Ulmer et al. (2023).

$$\mathrm{H}\big[p(\boldsymbol{\pi} \mid \boldsymbol{\alpha})\big] = \log \mathrm{B}(\boldsymbol{\alpha}) + (\alpha_0 - K)\psi(\alpha_0) - \sum_{k=1}^{K}(\alpha_k - 1)\psi(\alpha_k). \tag{2.76}$$

Unfortunately though, it should be noted that such data is often not available or in the first place, or cannot guarantee robustness against *other* kinds of unseen OOD data, of which infinite types exist in a real-valued feature space.[29]

Posterior Networks. When parameterizing Equation (2.68) instead of the Dirichlet prior, the neural networks now predicts the update N_k instead, and the prior parameters $\boldsymbol{\alpha}$ are typically set to be uniform. Nevertheless, we still need to gently guide the resulting Dirichlet posterior to attain its desired uncertainty behavior. Similar to prior networks, this can be done with an entropy regularizer (Sensoy et al., 2018) or additional training objective on OOD examples, including works that create synthetic OOD inputs using additional generative models (Sensoy et al., 2020; Hu et al., 2021). More interestingly, Charpentier et al. (2020); Stadler et al. (2021); Charpentier et al. (2022) use normalizing flows (Rezende and Mohamed, 2015) trained on the model's latent representations to compute the update N_k. By modeling the latent density, this allows us to update the uniform prior by a lot when the latent encoding is familiar, and leave the prior ignorance intact when it is not, and is therefore assigned a low probability by the normalizing flow.

2.2.4 Other Approaches

A number of other methods for UQ do not neatly fall into the categories we discussed so far. This includes for instance some works that see the layer-wise transformations happening inside a neural network as a dynamical system that can be modeled through neural stochastic differential equations (SDEs; Kong et al., 2020b; Wang et al., 2021c; Wang and Yao, 2021; Xu et al., 2022). By parameterizing the drift and diffusion terms of a SDE by neural networks, the diffusion network can be used to predict model uncertainty. Ma et al. (2023) parameterize a layer-wise mean and covariance instead, but do not embed these in a SDE. In a completely different approach, Hu et al. (2022) obtain a sequence of probabilities for a specific input from different model snapshots during training, and then quantify the uncertainty in the frequency

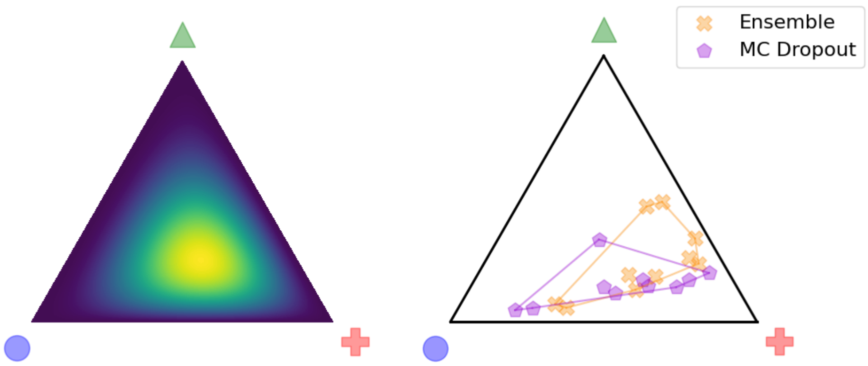
[29] The same applies to the synthetic OOD data in Chen et al. (2018); Shen et al. (2020); Sensoy et al. (2020).

domain after applying a discrete Fourier transform. Papernot and McDaniel (2018); Jiang et al. (2018) compare the output of a predictor to that of a simple nearest-neighbor classifier to quantify uncertainty, and Anirudh and Thiagarajan (2021) compare latent embeddings to a number of anchor points.

Direct Uncertainty Prediction. So far, we have treated uncertainty as something to be extracted from a model that, in general, is performing a different task, such as classification or regression. But what if we can just treat UQ as a supervised learning task, learning to predict an uncertainty score from an input? For instance, Geifman and El-Yaniv (2019) propose to add another prediction head to a model which predicts when the model should abstain from a potentially false output. The same option is instead parameterized as an additional class in a classification problem by Liu et al. (2019). Alternatively, the the confidence of a network can also be obtained from an independent network (Corbière et al., 2019, 2021; Luo et al., 2021; Fathullah et al., 2024; Liu et al., 2024b), which is also what Chapter 6 discusses in the context of LLMs. This model can also take the shape of a Gaussian process, as demonstrated by Qiu and Miikkulainen (2022).

Instead of setting up this additional model as a classifier, we can also employ a density estimator to derive the uncertainty of a target model, similar to posterior networks in Section 2.2.3. This again follows the idea that a density estimator would be able to indicate when a given test point lies outside of the known training distribution. As estimation of density can be achieved through Gaussian discriminant analysis on the latent representations (Mukhoti et al., 2021; Franchi et al., 2022), distances between latent features (Huang et al., 2021), kernel density estimators (Kotelevskii et al., 2022; Sun et al., 2024) or normalizing flows (Lahlou et al., 2023). Some of these methods are benchmarked by Postels et al. (2022), showing some sensitivity to distributional shifts nevertheless.

Credal Sets. Credal sets are based on the theory of imprecise probabilities (Boole, 1854; Keynes, 1921; Walley, 1991). The theory focuses on the idea that while there might a model that precisely describes a probability of interest, it may not be known, for instance due to vague, conflicting or scarce data (Caprio et al., 2023). One option to model this impreciseness is the use *credal sets*, which are sets of credible probability distributions. Like EDL methods in Section 2.2.3, they are defined on the probability simplex, but in contrast are not distributions, but convex sets instead. More intuitively, we can see a label y as a sample from the conditional



(a) Prior network prediction. (b) Credal sets from convex hulls.

Figure 2.10: Juxtaposition of a prior network and credal sets constructed from the convex hull of ensemble and MC dropout predictors.

distribution $y \sim P(y \mid \mathbf{x})$. Now, let the probability simplex for a classification problem with K classes be defined as

$$\Delta^{K-1} = \{\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_K)^{\mathrm{T}} \mid \lambda_k \geq 0, ||\,\boldsymbol{\lambda}\,||_1 = 1\} \subset \mathbb{R}^K, \quad (2.77)$$

and thus we can see that every $P(y \mid \mathbf{x}) \in \Delta^{K-1}$. A credal set $\mathcal{Q}$ is now a convex subset of this simplex, i.e. $\mathcal{Q} \subseteq \Delta^{K-1}$. As with evidential methods, ignorance about a prediction can be represented through including the whole simplex, so $\mathcal{Q} = \Delta^{K-1}$. Since the combination with neural models is still a nascent field of research, learning credal sets can be challenging. Existing ideas include self-supervised learning (Lienen and Hüllermeier, 2021a; Lienen et al., 2023), creating a convex hull around predictions produced by Bayesian methods such as ensembles (Mortier et al., 2022). We show an example of this for the second test point from the Iris dataset example from Section 2.2.3 in Figure 2.10. It is also possible to learn credal sets from Dirichlet networks when target *distributions* (instead of labels) are available (Javanmardi et al., 2024), using interval neural networks (which produce intervals over predictions and activations; Wang et al., 2024b). A more complex approach involves defining credal sets for priors and likelihood functions, from which credal sets of posterior distributions can be learned using variational inference (Caprio et al., 2023). In terms of uncertainty quantification, Mortier et al. (2022) develop several metrics to assess the calibration of credal predictors, and Hüllermeier et al. (2022); Sale et al. (2023b) investigate different uncertainty metrics. It should be mentioned that while a notion of volume of the credal sets appears as an intuitive metric (analogous to prediction set size), this intuition is flawed for multi-class classification problems (Sale et al., 2023b). In this regard, Hüllermeier et al. (2022) offer alternative metrics based class dominance (whether a certain class in more likely than all others for all the distributions in the credal set), which also allows to distinguish aleatoric from epistemic uncertainty.

2.3 Uncertainty in Natural Language Processing

Many of the approaches of uncertainty in the previous sections have also been applied to natural language processing, and we thus only mention some of the relevant works briefly: Calibration for instance has been investigated for classification (Desai and Durrett, 2020; Dan and Roth, 2021; Xiao et al., 2022; Ulmer et al., 2022b; Ahuja et al., 2022; Park and Caragea, 2022; Holm et al., 2023; Chen et al., 2023b; Zhu et al., 2023; Li et al., 2024c; Ye et al., 2024; Plaut et al., 2024). It has also been looked into in the context of generation tasks like language modeling (Zhu et al., 2023), machine translation (Wang et al., 2020b), and especially question-answering (Zhang et al., 2021c; Si et al., 2022, 2023; Lin et al., 2022a; Huang et al., 2023; Zhang et al.; Geng et al., 2023; Detommaso et al., 2024; Ulmer et al., 2024a). Conformal prediction has also been applied to NLP in various ways (see e.g. Campos et al. (2024) for a more comprehensive survey): These applications include natural language generation (Schuster et al., 2022; Ravfogel et al., 2023; Deutschmann et al., 2024; Ulmer et al., 2024c), prompt selection (Zollo et al., 2023), planning problems with LLMs (Ren et al., 2023), and behavioral alignment, i.e. the avoidance of toxic or otherwise undesired behaviors (Gui et al., 2024). Furthermore, some works have also sought out applications of evidential deep learning in NLP (Shen et al., 2020; He et al., 2023a), however with no application to language generation at the time of writing of this thesis.

Token- and Sequence-Level Uncertainty. Due to the sequentiality of language, uncertainty in NLP can be quantified on different scales. On the one hand, we might be interested in quantifying uncertainty on a (subword-)token level in order to e.g. identify mistranslations or factual errors. On the other hand, sequence-level uncertainties are of interest when the whole generation might be unreliable, or when we are trying to assess its usefulness for a downstream task. Similarly, there are potential applications to quantify uncertainty even on a paragraph-, document-, or dialogue-level. In order to now quantify the uncertainty on these scales, one might intuitively resort to the approaches for frequentist networks in Section 2.2.1, i.e. take the probability of the most likely token or the likelihood of a generated sequence as confidence. This runs into multiple problems: Due to the paraphrasticity of language (Section 2.1.3), a distribution over tokens might simply be uncertain due to the natural variability of language, not due to the uncertainty of the model.[30] Since we would like confidence scores to reflect some notion of correctness or reliability, using the likelihood of a generated sequence is also problematic; for one, token probabilities likely do not reflect confidence by themselves, but there is even a mismatch between the frequency of generated sequences compared to the (true) human distribution (Ott et al., 2018; LeBrun et al., 2022; Ji et al., 2023a), implying that sequence likelihoods are not even representative as the expected relative frequency of a generated sentence. This rules out their use to for instance reliably identify anomalous outputs. More importantly, there is no explicit inductive bias in modern architecture or training procedures that models the variability directly (Baan et al., 2024) or would push sequence likelihoods to reflect confidence per se (see for instance the results by Xue et al., 2024a; Becker and Soatto, 2024). While calibrating these likelihoods (Ulmer et al., 2024a; Xie et al.) or reweighing token probabilities in a sequence (Lin et al., 2024) can lead to some success, ECE results might also be misleading when comparing models with humans in a language context (Ilia and Aziz, 2024). Therefore, uncertainty on a sequence-level has instead been investigated by resampling generations (see next paragraph; Ott et al., 2018; Aina and Linzen, 2021). On a token-level, several approaches have emerged, for instance computing uncertainty given specific claims (Fadeeva et al., 2024), predicting the confidence based on the quantiles of the token distribution (Gupta et al., 2024), or training an additional prediction head (Kadavath et al., 2022). In order to compare a wide variety of different such uncertainty metrics, Huang et al. (2024) proposed the use of *rank calibration*, i.e. testing whether higher certainty indeed implies higher generation quality. Some works also exists that quantify uncertainty for long texts, for instance based on the entailment probabilities of segments (Zhang et al., 2024a), and Sicilia et al. (2024) model the uncertainty inherent in long conversations (but therefore not the uncertainty of the model processing the conversation itself).

Self-consistency, Prompt Ensembling and Output Diversity. While there has been some research over the years into Bayesian methods (Xiao et al., 2020; Malinin and Gales, 2021; Gidiotis and Tsoumakas, 2022; Xiong et al., 2023), these have become less

[30] Neural models also have been show to be ill-calibrated towards the human word distribution, see Liu et al., 2024a; Ilia and Aziz, 2024.

applicable in the era of large language models due to their sheer size.[31] Therefore, a number of works ensemble predictions for the same input (also referred to as *self-consistency*; Wang et al., 2023b; Manakul et al., 2023; Chen and Mueller, 2023; Li et al., 2024b), from the same prompt with different pieces of additional information (Hou et al., 2023a), or from different prompts altogether (Li et al., 2023b; Hou et al., 2023b; Pitis et al., 2023; Gao et al., 2024b) instead of predictions from different parameter sets. The intuition remains similar to Bayesian methods in Section 2.2.2: If similar prompts for the same input produce vastly different predictions, the network must be uncertain. We can therefore interpret prompt ensembling techniques as evaluating a predictive distribution over distribution of prompts $p(\boldsymbol{\rho})$ and in-context samples $p(\mathcal{C})$:

$$\mathbb{E}_{p(\boldsymbol{\rho},\mathcal{C})}\big[p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\rho}, \mathcal{C})\big] = \iint p(\mathbf{y} \mid \boldsymbol{\theta}, \mathbf{x}, \boldsymbol{\rho}, \mathcal{C}) p(\boldsymbol{\rho}) p(\mathcal{C}) \mathrm{d}\boldsymbol{\rho}\, \mathrm{d}\mathcal{C}. \tag{2.78}$$

Any disagreement in responses however can also be influenced by the generation hyperparameters, and thus this method does not admit a clean distinction between aleatoric and epistemic uncertainty like in Equation (4.4).[32] Furthermore, Ling et al. (2024) investigate how the choice of in-context samples can also induce additional uncertainty into the LLMs generation. Kuhn et al. (2023) base their idea of *semantic entropy* on a similar intuition: Trough the use of a bi-directional entailment classifier, generations are clustered by meaning.[33] Instead of the Shannon entropy over classes in Equation (4.2), we evaluate entropy over all the sequences $\mathbf{s}$ given some $\mathbb{M}$ out of M clustered meaning classes:

$$\begin{aligned}
\mathrm{SE}(\mathbf{x}) &= -\sum_{m=1}^{M} p(\mathbb{M}_m \mid \mathbf{x}) \log p(\mathbb{M}_m \mid \mathbf{x}) && (2.79)\\
&= -\sum_{m=1}^{M} \Big(\sum_{\mathbf{s}\in\mathbb{M}_m} p(\mathbf{s} \mid \mathbf{x}) \Big) \log \Big(\sum_{\mathbf{s}\in\mathbb{M}_m} p(\mathbf{s} \mid \mathbf{x}) \Big) && (2.80)\\
&\approx -\frac{1}{M} \sum_{m=1}^{M} \log \Big(\sum_{\mathbf{s}\in\mathbb{M}_m} p(\mathbf{s} \mid \mathbf{x}) \Big), && (2.81)
\end{aligned}$$

where the last step is obtained through Monte Carlo integration. Aichberger et al. (2024) improve on this estimator by producing

[31] This comes with the exception of methods like Yang et al., 2023; Onal et al.. Besides, Papamarkou et al. (2024) sketch avenues with which Bayesian methods can still provide advantages in the age of large-scale methods.

[32] In contrast to the claims of Hou et al., 2023a.

[33] The idea is that if the classifier indicates that a generation implies another and vice versa, they must (ought to) be equivalent.

more variable generations through targeted token substitutions. Instead of computing the entropy over hard meaning clusters, Nikitin et al. (2024) propose to instead compute the entropy using semantic kernels that measure the similarity in meaning between model responses, replacing hard clusters.

Verbalized Uncertainty. Originating from works like T5 (Raffel et al., 2020), natural language has become a general interface for modern NLP models. This refers both to embedding other, traditionally non-generative tasks such as sequence classification into a sequence-to-sequence task, but also to users increasingly interacting with language models through prompting. Mielke et al. (2022) already demonstrated that pre-trained models could be finetuned to express different levels of uncertainty in words. This however required finetuning on human-annotated data, while modern approaches simply prompt the LLM to express its uncertainty in words (Kadavath et al., 2022; Xiong et al., 2023; Tian et al., 2023; Chen et al., 2023a), often through percentage values ("Confidence: 96 %") or confidence expressions ("Confidence: Very high"), which are then mapped back onto numerical values for evaluation purposes. Tian et al. (2023) for instance find that through the combination of suitable prompts and temperature-scaling, the calibration error of such methods can be noticeably reduced. However, they also find that the distributions of confidence expressions are highly skewed—while it does differ between datasets, the tested GPT models (GPT-3.5 and GPT-4) tend to mostly confident expressions, likely due to the unequal usage of these terms in their training data. This finding is corroborated by Yona et al. (2024); Singh et al. (2024a); Krause et al. (2023), indicating that LLMs always generate decisive answer even for uncertain questions, and that this is challenging to change through prompting alone. When results are strong, this might coincide with cases in which the dataset is too easy and the skewed confidence expression distribution actually conforms to the results (as for instance for TriviaQA in Ulmer et al., 2024a; Xue et al., 2024a). Lin et al. (2022a) also finetune an LLM to verbalize its uncertainty, but do so on automatically generated confidence targets that are obtained by checking the model's performance on some sub-category of a task, like different question types for mathematical reasoning. A similar approach is taken by Zhang et al. (2023a), finetuning them to admit their uncertainty for incorrect answers. In the case of Kadavath et al. (2022), the LLM is simply asked directly whether its answer was true or false. Band et al. (2024) finetune verbalized uncertainty from a Bayesian decision-making standpoint, increasing factuality. Zhou et al. (2023) investigate the general use of linguistic confidence expressions in LLMs, and show that accuracy can be influenced through the use of such expressions in the prompt.

Uncertainty for Black-box Models. The commercialization of LLM-based chatbots such as ChatGPT (OpenAI, 2022) also created a trend of black-box models, which are shielded by an API. As such, any UQ method has to do without any access to model latent representations, logits or output probabilities. The question of whether and how uncertainty can be estimated from text generations alone therefore also has become an active area of research. Such approaches include predicting confidence directly from the generated text using an auxiliary model (Chapter 6; Ulmer et al., 2024a), verbalized uncertainty methods from the previous paragraph, or comparing the similarity of generations given the same input (Lin et al., 2023). Su et al. (2024) further show that LLM predictions can be conformalized even without access to the probabilities through repeated sampling and word frequencies analysis alone.

Reward Modeling. As part of the contemporary language model pipeline, models are first pre-trained on large amounts of text using a language modeling objective (Devlin et al., 2019; Radford et al., 2019), then finetuned on a number of instructions, and finally undergo a step that aims to align their behavior with general human values (Ouyang et al., 2022). This last step is often performed using reinforcement learning from human feedback (RLHF; Christiano et al., 2017; Stiennon et al., 2020). This involves the use of a trained reward model, that predicts the quality of a generation based on human preference data. While it has been found that this step can hurt model calibration (Zhu et al., 2023), the reward modeling itself has also been characterized as brittle, and thus a number of works have proposed Bayesian approaches to the target model finetuning or the reward model to increase robustness (Zhai et al., 2024; Yang et al., 2024; Zhang et al., 2024b,c).

Human Label Variation. Compared to other input modalities, the variability, ambiguity and underspecification of language (Section 2.1.3) calls the validity of a single ground truth for training into question. Indeed, there have been several calls to embrace this diversity for classification (Basile et al., 2021; Plank, 2022; Baan et al., 2022; Gruber et al., 2024) and language generation tasks (Baan et al., 2023). Importantly, this opens up new avenues for better modeling and representing of the uncertainty in the underlying data (Nie et al., 2020; Zhou et al., 2022; Uma et al., 2021; Davani et al., 2022; Wu et al., 2023a), modeling annotators

(Deng et al., 2023), and to learn from fewer instances (Gruber et al., 2024). Training on single labels or references has for instance been hypothesized to cause the miscalibration of neural models to human language variability (Giulianelli et al., 2023; Ilia and Aziz, 2024), and to potentially be responsible for the inadequacy of greedy decoding in natural language generation (Eikema and Aziz, 2020; Eikema, 2024). The variation of labels should therefore be reframed as an opportunity, as it for instance also allows to more easily learn second-order predictors like evidential neural networks or credal sets (Javanmardi et al., 2024).

2.4 Uncertainty & Trust

Even though we have already discussed several applications of uncertainty quantification in Section 1.2 in the first chapter, it is useful to zoom in on the aspect of trust, why it matters, and how quantifying the uncertainty of a ML system can help. The reason for this is the following: The main promise of machine learning algorithm lies in its ability to analyze and process large swaths of data, identifying potential patterns that remain elusive for even the most astute humans. As such, it promises to either replace or support human decision-makers. However, even if a part of the deliberation for a decision is taken over by a machine, people are the ones that remain affected by it. This is true for all the examples of decision support including for medical staff, self-driving cars or automated translation systems. Trust is the social mechanism that governs this relationship, and is a necessary requirement for it to have a positive effects. If trust is not present, we run the risk of alienating the people affected, leading to them ignoring the automation and thus foregoing any benefits, or even creating negative consequences. Indeed, Inie (2024) finds in a diverse survey that participants perceive AI systems as less trustworthy when problems they are trying to solve or the models themselves are complex, and when no human expert is in the loop.

Jacovi et al. (2021) formalize this dynamic using notions of interpersonal trust from sociology. They thereby define two roles: The trustor (i.e. the person trusting someone) and the trustee (i.e. the person being trusted). In order to make this distinction clearer in our context, we will notate these roles by trustor 👱 and trustee 🤖. They employ the following definition of interpersonal trust:

Definition 1 (Interpersonal Trust; Mayer et al., 1995)**.** If a trustor 👱 believes that a trustee 🤖 will act in their best interest and accepts vulnerability to the trustee 🤖 's actions, then the trustor 👱 trusts the trustee 🤖 .

The authors admit that this definition is somewhat simplistic: AI systems are not people, and as such, terms such as *reliance* (i.e., the trust put into an object) might be more applicable (Baier, 1986). However, users often show tendencies to anthropomorphize AI systems (Miller, 2019; Jacovi and Goldberg, 2021). And thus, we can use a variation of Definition 1 to define human-AI trust. Jacovi et al. here use the notion of contract between the trustor 👱 and the trustee 🤖 , which in the human-AI case has to be explicit instead of implicit. Such contracts define certain properties or behaviors that model is expected to uphold. This can include things as for instance robustness, fairness w.r.t. certain group in the datasets, or interpretability and finally leads us to the definition of human-AI trust:

Definition 2 (Human-AI Trust; Jacovi et al., 2021)**.** A trustee 🤖 in the form of an AI model is trustworthy if it is capable of maintaining a specific contract with the trustor 👱 .

Jacovi et al. further distinguish two kinds of trust: *Intrinsic trust*, when the decision process of the trustee 🤖 is observable and matches the trustor 👱 's own priors. This is possible in a decision tree, but very hard for neural networks, as their size can obscure the decision process. Therefore, we focus here on *extrinsic trust*, which is built by observing symptoms of a trustworthy model. A symptom of a trustworthiness can for example be its (consistent) performance of the trustee 🤖 model, as for instance explored by Yin et al. (2019); Rechkemmer and Yin (2022). While the above assumed the trustor 👱 to be human and the trustee 🤖 to be an AI system, recent work has also started exploring whether AI systems can exhibit human trust behaviors (Xie et al., 2024).

It can be argued that one such tool for building extrinsic trust can be uncertainty quantification methods: Using them, the trustee 🤖 can communicated how much weight should be assigned to its predictions, and when they are better to be ignored. Further, explicit contracts like in Definition 2 can be formed by providing model cards that for instance report the calibration of a model on specific datasets. Overall, Liao and Sundar (2022) describe that such trust in automation is not inherent, and that additional care has to be put into how to design the trust cues for an end user. For this reason, we will discuss ways of communicating uncertainty next.

2.5 Communicating Uncertainty

Understanding the usefulness of a model can be challenging for laypeople and experts alike. Even when possessing technical domain knowledge, NLP practitioners for instance struggle to select the best encoder model for a task (Bassignana et al., 2022). Even accuracy scores or other performance metrics can be hard to interpret, especially when they may unknowingly degrade under distributional shift in an application. The previous sections have demonstrated the diversity of ways in which uncertainty is measured, often requiring knowledge about the model, methods or entire schools of thought (as in the frequentist vs. the Bayesian example). In practice, requiring such knowledge from laypeople is unrealistic; furthermore, the interpretation of such measure is also influenced by human numeracy (Zikmund-Fisher et al., 2007; Galesic and Garcia-Retamero, 2010) and cognitive biases (Reyna and Brainerd, 2008; Daniel, 2017; Spiegelhalter, 2017). Therefore, Bhatt et al. (2021) advocate that in practice, uncertainty measure should be tailored to and tested with the different stakeholders they are targeted towards. This includes an arsenal of ways such as communicating numerical values, graphical means or the verbalized uncertainty from Section 2.3. However, the best way of communicating uncertainty in an NLP context remains application-dependent and underexplored. One promising avenue is the verbalized uncertainty in Section 2.3, although this approach at its current stage remains quite simplistic: Usually, uncertainties are communicated as percentage values or values on a discrete scale, instead of making use of the rich variety in human uncertainty expressions (Section 2.1.4).

Effects of Communicating Uncertainty. Some works have investigated how communicated uncertainty influences the trust of human users. For instance, Zhang et al. (2020d) show how displaying confidence scores can help to calibrate people's trust in a model, but that it may not necessarily improve the outcomes of AI-assisted decision making, whereas Kim et al. (2024b) find a positive effect on accuracy in a human study with LLMs. Paradoxically, Vodrahalli et al. (2022) show how these outcomes can be improved even when the underlying confidence scores are not calibrated. In another experiment with human participants, Dhuliawala et al. (2023) showcase how misleading uncertainty can produce lose-lose situations. In their study, they quantify human trust in uncertainty estimates through monetary bets on a model's answers in a question-answering task. They find two things in the face of unreliable uncertainty estimates: Firstly, a smaller overall pay-off for the participants and a loss of trust in the system, both caused due to or signified by more conservative bets. In general, it should also be noted that notions like trust are notoriously hard to isolate in human experiments, and that any stated results also presuppose a specific model between model predictions and their influence on human decision-making.

2.6 Applications of Uncertainty

Previous sections have focused on characterizing and quantifying uncertainty that one encounters in machine learning and natural language processing. This is not a purely intellectual quest, and we have already touched on some potential use-cases in Section 1.2. There is exists a trove of research works on several downstream applications that uncertainty quantification can be used for, a (non-exhaustive) list of which we present here.

Fairness. Algorithmic fairness has recently increased in popularity as a field that studies systematic biases and mitigation strategies in AI algorithms (Pessach and Shmueli, 2023). In this regard, some works have researched Bayesian treatments of fairness metrics (Ji et al., 2020; Kuzucu et al., 2023; Barrainkua et al., 2024). Others have argued that uncertainty can be a source of unfairness (Singh et al., 2021; Ali et al., 2021; Tahir et al., 2023; Wang et al., 2024a; Cooper et al., 2024) and propose its quantification as a way to reduce bias during training (Stone et al., 2022). The relationship between debiasing techniques and UQ has further been investigated by Kuzmin et al. (2023).

Error Detection. Since uncertainty estimates are usually hard to evaluate due to the lack of ground truth, and thus error detection is both a downstream application as well as an evaluation strategy. The intuition lies in the fact that predictions with higher uncertainty should assumed to be more likely to be wrong. Examples for this are for instance the works of Kong et al. (2020a); Ashukha et al. (2020); Vazhentsev et al. (2022); Thuy and Benoit (2023); Vazhentsev et al. (2023), among many others. In the context of LLMs, uncertainty quantification has also been applied specifically to hallucination detection (Xiao and Wang, 2021; Manakul et al., 2023; Zhang et al., 2023b; Band et al., 2024; Detommaso et al., 2024).

Out-of-distribution Detection. Out-of-distribution detection follows a similar logic as error detection. As inputs different from the training data of a model should could lead to unexpected predictions since the model is underspecified on them (i.e. different models that fit the training data will create disagreeing predictions on unseen data; D'Amour et al., 2022), we want the model to be generally more uncertain about its prediction. In contrast to error detection however, this applications focuses on model uncertainty, since errors can be caused by high model uncertainty or inherent difficulty alike. This assumptions has been shown to be formally incorrect for some simple ReLU networks (Section 4.1; Hein et al., 2019; Ulmer and Cinà, 2021), and the ability of uncertainty to detect OOD inputs has been investigated in a larger number of works (see, among many others, DeVries and Taylor, 2018; Snoek et al., 2019; Liu et al., 2023; Ulmer et al., 2020; Kong et al., 2020a; Stadler et al., 2021; Arora et al., 2021; Ulmer et al., 2022b; Uppal et al., 2024).

Conditional Computation. Uncertainty can also be used as a signal to switch the intended way of processing for an input, which can be motivated by cognitive reasons (e.g. based on system 1 and system 2 in humans; Daniel, 2017), boosting performance (Gerych et al., 2024) or to improve efficiency (Schuster et al., 2022; Varshney and Baral, 2022). Gerych et al. (2024) for instance use confidence scores to route inputs to a pool of models to find the best-performing one, and Zheng et al. (2019) use uncertainty to determine the right module from a mixture of experts. In NLG, van der Poel et al. (2022) use mutual information to switch the decoding algorithm, and Xiao and Wang (2021) adapt beam search based on uncertainty in order to alleviate hallucinations. Another usage of uncertainty enables the early exciting from a model, i.e. where not all layers of a deep learning model are used (Schuster et al., 2022; Fei et al., 2022; Bajpai and Hanawal, 2024). Lastly, uncertainty has also been utilized in *model cascades*, where we try to select one of a pool of increasingly-sized model based on the difficulty of an input (Teerapittayanon et al., 2016; Varshney and Baral, 2022; Jitkrittum et al., 2024; Gupta et al., 2024).

Active Learning. Active learning describes a field of machine learning in which an algorithm selects unlabeled instances that are given to a human for labeling, and are subsequently added to the algorithm's training data (Settles, 2009). The use of uncertainty measures for this purpose has long predated deep neural networks (e.g. Lewis and Gale, 1994; Lewis and Catlett, 1994; Scheffer et al., 2001), and has found many applications since their revival (Ren et al., 2021; Zhang et al., 2022c). When using uncertainty to identify samples of interest, there also exists a colorful bouquet of approaches: Frequentist methods usually rely on some measure of model confidence (Wang and Shang, 2014; Matiz and Barner, 2019; Ebrahimi et al., 2020; Zhang and Plank, 2021; Wang and Plank, 2023), Bayesian methods quantify metrics such as mutual information (Gal et al., 2017b; Kirsch et al., 2019; Kim et al., 2021; Kirsch and Gal, 2022; Smith et al., 2023) and evidential methods utilize distributional uncertainty (Zhu et al., 2021; Park et al., 2022; Hemmer et al., 2022).

Requesting Human Oversight. Active learning is a specific case of human-in-the-loop problems that is focused on resource-efficient data labeling, but can be seen as just one instance of a class of applications in which human oversight or intervention is requested upon uncertainty. Other examples include for for instance planning problems in reinforcement learning (Singi et al., 2023), industrial applications (Treiss et al., 2021), clarifying uncertain parts in image segmentation for remote sensing (García Rodríguez et al., 2020), text moderation (Andersen and Maalej, 2022; Andersen and Zukunft, 2022), and co-annotation of data (Li et al., 2023a). In general, these applications promise to alleviate the workload that would be otherwise assigned to human experts, and only request their assistance in the case of difficult inputs.

2.7 Summary

This chapter has given a fairly comprehensive account of uncertainty and its relevant concepts, definitions, methods and applications for deep learning and natural language processing. It has provided an overview over the different definitions of uncertainty in statistics, i.e. the frequentist and Bayesian viewpoints, and how uncertainty in linguistics plays a layered role as an inherent feature of language on the one side, and a tool for communication of one's world state on the other. These different notions crystallize in their applications to neural networks: Statistical uncertainties permeate model training and inference, and linguistic uncertainties influence the processing of natural language inputs. Not only is the quantification of these uncertainties challenging and methods to do so are multifarious, but the adequate communication of uncertainty is equally difficult. This last step is pivotal to enable human-AI collaboration, in which trust relationships are formed between users and their silicate collaborators. As with human relationships, this trust can be built but also lost, which suggests more research is needed to understand this dynamic better.

3 Addressing Uncertainty in Experimental Design

“*When you run an experiment, you take notes, think for a while, then publish your results. If you don’t publish, nobody will learn from your experience. The whole idea is to save other from repeating what you’ve done.*”

—Clifford Stoll in *The Cuckoo’s Egg: Tracking a Spy Through the Maze of Computer Espionage.*

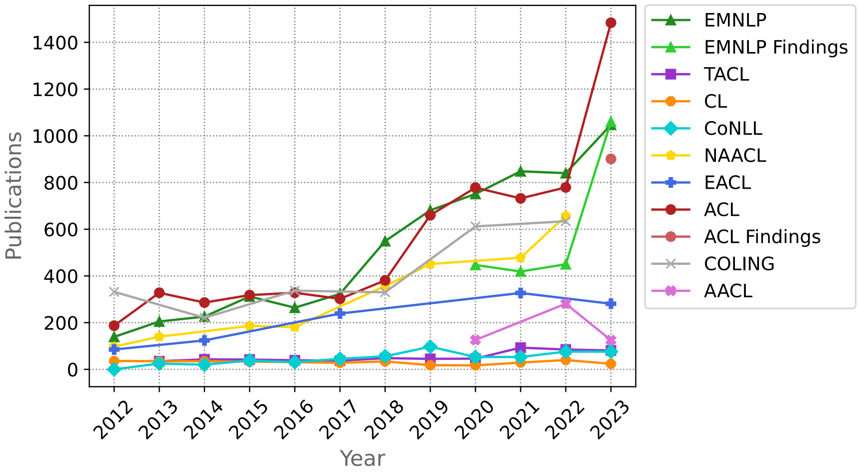


Figure 3.1: Published papers at NLP venues. Gaps are due to some venues not producing proceedings in any given year. Notably, this plot does not include NLP papers published at venues such as NeurIPS, ICML, or ICLR.

Before returning to the uncertainty in NLP *models*, we will first engage in another, wider perspective on where uncertainty hides in the NLP pipeline. DL in general, and NLP in its current form, are largely empirical sciences: We obtain new knowledge by forming hypotheses, running experiments, and then analyzing results to come to a conclusion about our initial suppositions. In the the last decade or so, this field has ballooned in size: In Figure 3.1, we show the number of published conference papers in NLP venues since 2012, which has more than quadrupled.

While such growth is remarkable, it comes at a cost: Akin to concerns in other disciplines (John et al., 2012; Jensen et al., 2021), several authors have noted major obstacles to reproducibility (Gundersen and Kjensmo, 2018; Belz et al., 2021) and a lack of hypothesis testing (Marie et al., 2021) or published results not carrying over to different experimental setups, for instance in text generation (Gehrmann et al., 2022) and with respect to new model architectures (Narang et al., 2021). Others have questioned commonly-accepted experimental protocols (Gorman and Bedrick, 2019; Søgaard et al., 2021; Bouthillier et al., 2021; van der Goot, 2021) as well as the (negative) impacts of research on society (Hovy and Spruit, 2016; Mohamed et al., 2020; Bender et al., 2021; Birhane et al., 2022) and environment (Strubell et al., 2019; Schwartz et al., 2020; Henderson et al., 2020). Lastly, the adoption of large language models that are also possibly closed-source have exacerbated problems about experimental protocols further (Mizrahi et al., 2024; Balloccu et al., 2024). These problems have not gone unnoticed—many of the mentioned works have proposed a cornucopia of solutions. In a quickly-moving environment however, keeping track and implementing these proposals becomes challenging.

This chapter addresses these issue in two ways: On the one hand, open issues in reproducibility and replicability are woven together into a cohesive set of guidelines for gathering stronger experimental evidence, that can be implemented with reasonable effort and which are discussed in Section 3.1. On the other hand, we zoom into the question of hypothesis testing (Section 3.2), with a specific focus on the almost stochastic order test (ASO; del Barrio et al., 2018a; Dror et al., 2019) in Section 3.2.1 and its application to question-answering with LLMs in Section 3.2.2. The core thesis of this chapter is that increased efforts in reproducibility and replicability are intricately linked to the question of uncertainty in empirical research: For example, transparent and diligent data curation enables better modeling of uncertainty (referring to the discussion on language paraphrasticity in Section 2.1.3 and human label variation in Section 2.3), and a more rigorous experimental protocol and statistical hypothesis testing can help to unveil the uncertainty lingering in results, aiding the development of better methods and bringing more clarity to the research landscape. Therefore, we build these ideas up from the scientific method and show their implementation in the experimental pipeline.

3.1 Experimental Standards for NLP

The following work is based on Ulmer et al. (2022a).

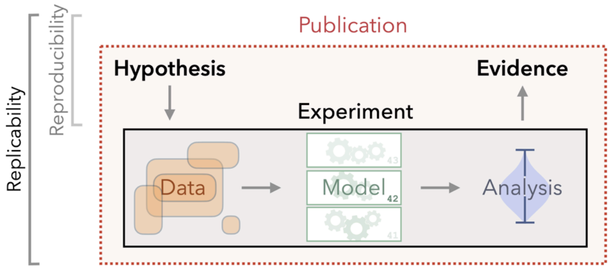


Figure 3.2: Schematic representation of the scientific method in Deep Learning. After forming hypotheses, we conduct our experiments by modeling some data of interest and analyzing the results to obtain some evidence to support or reject our initial assumptions. While reproducibility entails the *reproduction* of evidence based on the hypotheses and a description of the experiments, replicability refers to a step-by-step copy of the pipeline using the original data, model, and analyses.

The Scientific Method. Knowledge can be obtained through several ways including theory building, qualitative methods, and empirical research (Kuhn, 1970; Simon, 1995). Here, we focus on the latter aspect, in which (exploratory) analyses lead to falsifiable hypotheses that can be tested and iterated upon (Popper, 1934).[34] This process requires that *anyone* must be able to back or dispute these hypotheses in the light of new evidence.

In the following, we focus on the evidence-based evaluation of hypotheses and how to ensure the scientific soundness of the experiments which gave rise to the original empirical evidence, with a focus on *replicability* and *reproducibility*. In computational literature, one term requires access to the original code and data in order to re-run experiments exactly, while the other requires sufficient information in order to reproduce the original findings even in the absence of code and original data (see also Figure 3.2).[35]

[34] While such hypothesis-driven science is not always applicable or possible (Carroll, 2019), it is a strong common denominator that encompasses most empirical ML research.

[35] Strikingly, these central terms already lack agreed-upon definitions (Peng, 2011; Fokkens et al., 2013; Liberman, 2015; Cohen et al., 2018), however we follow the prevailing definitions in the NLP community (Drummond, 2009; Dodge and Smith, 2020) as the underlying ideas are equivalent.

Replicability. Within DL, we take replicability to mean the (near-)exact replication of prior reported evidence. In a computational environment, access to the same data, code and tooling should be sufficient to generate prior results. However, many factors, such as hardware differences, make exact replication difficult to achieve. Nonetheless, we regard experiments to be replicable if a practitioner is able to re-run them to produce the same evidence within a small margin of error dependent on the environment, without the need to approximate or guess experimental details.

Reproducibility. In comparison, we take reproducibility to mean the availability of all necessary and sufficient information such that an experiment's findings can be independently reaffirmed when the same research question is asked. As discussed later, the availability of all components for replicability is rare—even in a computational setting. An experiment then is reproducible if anyone with access to the publication is able to re-identify the original evidence, i.e. exact results differing, but patterns across experiments being equivalent. This is illustrated by Figure 3.2, where replicability involves access to all data, modeling and analysis steps, whereas reproducibility only involves knowledge of the hypotheses, a description of the experiments, as well as their results.

We assume that the practitioner aims to follow these principles in order to find answers to a well-motivated research question by gathering the strongest possible evidence for or against their hypotheses. The guidelines in the following sections therefore aim to model or reduce uncertainty in each step of the experimental pipeline through enhancing its reproducibility and / or replicability.

3.1.1 Data

Frequently, it is claimed that a model solves a particular cognitive task, however in reality it merely scores higher than others on some specific dataset according to some predefined metric (Schlangen, 2021). Of course, the broader goal is to improve systems more generally by using individual datasets as proxies. Admitting that our experiments cover only a small slice of the real-world sample space will help more transparently measure progress towards this goal. In light of these limitations and as there will always be private or otherwise unavailable datasets which violate replicability, a practitioner must ask themselves: *Which key information about the data must be known in order to reproduce an experiment's findings?* In this section we define requirements for putting this question into practice during dataset creation and usage such that anyone can draw the appropriate conclusions from a published experiment.

Choice of Dataset. The choice of dataset arises from the need to answer a specific research question within the limits of the available resources. Such answers typically come in the form of comparisons between different experimental setups while using the equivalent data and evaluation metrics. Using a publicly available, well-documented dataset will likely yield more comparable work, and thus stronger evidence. In absence of public data, creating a new dataset according to guidelines which closely follow prior work can also allow for useful comparisons. Should the research question be entirely unexplored, creating a new dataset will be necessary. In any case, the data itself must contain the information necessary to generate evidence for the researcher's hypothesis. For example, a model for a classification task will not be learnable unless there are distinguishing characteristics between data points and consistent labels for evaluation. Therefore, an exploratory data analysis is recommended for assessing data quality and anticipating problems with the research setup. Simple baseline methods such as regression analyses or simply manually verifying random samples of the data may provide indications regarding the suitability and difficulty of the task and associated dataset (Kreutzer et al., 2022). On the flip side, a lower-quality dataset runs the danger of introducing noise and therefore aleatoric uncertainty into the dataset (Baan et al., 2023).

Metadata. At a higher level, data sheets and statements (Gebru et al., 2021; Bender and Friedman, 2018) aim to standardize metadata for dataset authorship in order to inform future users about assumptions and potential biases during all levels of data collection and annotation—including the research design (Hovy and Prabhumoye, 2021). Simultaneously, they encourage reflection on whether the authors are adhering to their own guidelines (Waseem et al., 2021). Generally, higher-level documentation should aim to capture the dataset's *representativeness* with respect to the global population. This is especially crucial for "high-stakes" environments in which subpopulations may be disadvantaged due to biases during data collection and annotation (He et al., 2019; Sap et al., 2022). Even in lower-stake scenarios, a model trained on only a subset of the global data distribution can have inconsistent behavior when applied to a different target data distribution and display high model uncertainty (D'Amour et al., 2022; Koh et al., 2021). For instance, domain differences have a noticeable impact on model performance (White and Cotterell, 2021; Ramesh Kashyap et al., 2021). Increased data diversity can improve the ability of models to generalize to new domains and languages (Benjamin, 2018), however diversity is difficult to quantify (Gong et al., 2019) and full coverage is unachievable. This highlights the importance of documenting representativeness in order to ensure reproducibility—even in absence of the original data. For replicability using the original data, further considerations include long-term storage and versioning, as to ensure equal comparisons in future work.

Instance Annotation. Achieving high data quality requires that the data must be accurate and relevant for the task to enable effective learning (Pustejovsky and Stubbs, 2012; Tseng et al., 2020) and reliable evaluation (Bowman and Dahl, 2021; Basile et al., 2021). Since most datasets involve human annotation, a careful annotation design is crucial (Pustejovsky and Stubbs, 2012; Paun et al., 2022). Ambiguity in natural language poses inherent challenges and disagreement is genuine (see Sections 2.1.3 and 2.3 or Basile et al., 2021; Specia, 2021; Uma et al., 2021; Plank, 2022). As insights into the annotation process are valuable, yet often inaccessible, we recommend to release datasets with individual-coder annotations, as also put forward by Basile et al. (2021); Prabhakaran et al. (2021); Plank (2022) and to complement data with insights like statistics on inter-annotator coding (Paun et al., 2022), e.g., over time (Braggaar and van der Goot, 2021), or coder uncertainty (Bassignana and Plank, 2022). When creating new datasets such information strengthens the reproducibility of future findings, as they transparently communicate the inherent variability instead of obscuring it. Furthermore, this opens up new avenues to model distributions instead of single gold labels to more accurately reflect uncertainty (Javanmardi et al., 2024; Gruber et al., 2024) or modeling single annotators (Deng et al., 2023).

Pre-processing. Given a well-constructed or well-chosen dataset, the first step of an experimental setup will be the process by which a model takes in the data. This must be well documented or replicated—most easily by publishing the associated code—as perceivably tiny pre-processing choices can lead to huge accuracy discrepancies (Fokkens et al., 2013) and influences model uncertainty during inference.[36] Typically, this involves decisions such as sentence segmentation, tokenization and normalization. In general,

[36] One could for instance imagine a case where data uncertainty is created by not removing certain characters like rare symbols or fragments of code, or increasing model uncertainty through suboptimal tokenization of a language, for instance through another language's or multilingual tokenizer (Rust et al., 2021).

the data setup pipeline should ensure that a model "observes" the same kind of data across comparisons. Next, the dataset must be split into representative subsamples which should only be used for their intended purpose, i.e. model training, tuning and evaluation (see Section 3.1.3). In order to support claims about the generality of the results, it is necessary to use a test split without overlap with other splits. Alternatively, a tuning / test set could consist of data that is completely foreign to the original dataset (Ye et al., 2021), ideally even multiple sets (Bouthillier et al., 2021), which is also essential when trying to quantify any model uncertainty in the face of distributional drifts (see Section 4.2). It should be noted that even separate, static test splits are prone to unconscious "overfitting", if they have been in use for a longer period of time, as people aim to beat a particular benchmark (Gorman and Bedrick, 2019). If a large variety of resources are not available, it is also possible to construct challenging test sets from existing data (Ribeiro et al., 2020; Kiela et al., 2021; Søgaard et al., 2021). Finally, the metrics by which models are evaluated should be consistent across experiments and thus benefit from standardized evaluation code (Dehghani et al., 2021). For some tasks, metrics may be driven by community standards and are well-defined (e.g. classification accuracy). In other cases, approximations must stand in for human judgment (e.g. in machine translation). In either case—but especially in the latter—dataset authors should inform users about desirable performance characteristics and recommended metrics.

Appropriate Conclusions. The results a model achieves on a given data setup should first and foremost be taken as just that. Appropriate, broader conclusions can be drawn using this evidence provided that biases or incompleteness of the data are addressed (e.g., results only being applicable to a subpopulation). Even with statistical tests for the significance of comparisons, properties such as the size of the dataset and the distributional characteristics of the evaluation metric may influence the statistical power of any evidence gained from experiments (Card et al., 2020). In experiments with large models, practitioners might decide to only run the model on a subset of the data. But again, such a sample might not be powerful enough and not enable fair comparisons with other models (Balloccu et al., 2024). It is therefore important to keep in mind that in order to claim the reliability of the obtained evidence, for example, larger performance differences are necessary on less data than what might suffice for a large dataset, or across multiple comparisons (see Section 3.1.3). Finally, a practitioner should be aware that a model's ability to achieve high scores on a certain dataset may not be directly attributable to its capability of simulating a cognitive ability, but rather due to spurious correlations in the input (Ilyas et al., 2019; Schlangen, 2021; Nagarajan et al., 2021). By for instance only exposing models to a subset of features that should be inadequate to solve the task, we can sometimes detect when they take unexpected shortcuts (Fokkens et al., 2013; Xenos et al., 2023). Communicating the limits of the data helps future work in reproducing prior findings more accurately.

Best Practices: **Data**

- ◇ Consider dataset & experimental limitations (Schlangen, 2021);
- ◇ Document task adequacy, representativeness and pre-processing (Bender and Friedman, 2018);
- ◇ Split the data such as to avoid spurious correlations;
- ◇ Publish the dataset accessibly & indicate changes;
- ⋆ Perform exploratory data analyses to ensure task adequacy (Kreutzer et al., 2022);
- ⋆ Publish the dataset with individual-coder annotations;
- ⋆ Consider the dataset's statistical power (Card et al., 2020).

3.1.2 Codebase & Models

The NLP community has historically taken pride in promoting open access to papers, data, code, and documentation, but some have also noted room for improvement (Wieling et al., 2018; Belz et al., 2021). The benefit of such a repository is in its ability to enable direct *replication*, helping to reduce uncertainty in modeling when building upon others work. In DL however, full datasets can be large and impractical to share. Due to their importance however, it is essential to carefully consider how one can share the data with researchers in the future. Therefore, repositories for long-term data storage backed by public institutions should be preferred (e.g. LINDAT / CLARIN by Váradi et al., 2008). Nevertheless, practitioners often can not distribute data due to privacy, legal, or storage reasons. In such cases, practitioners must instead carefully consider how to distribute data and tools to allow future research to produce accurate replications of the original data (Zong et al., 2020).

Hyperparameter Search. Hyperparameter tuning strategies remain an open area of research (e.g. Bischl et al., 2023), but are central to the replication of contemporary models. Well-chosen hyperparameters promote stability in model predictions, while illchosen parameters induce additional additional uncertainty.[37] The following rules of thumb exist: Grid search or Bayesian optimization can be applied if few parameters can be searched exhaustively under the computation budget. Otherwise, random search is preferred, as it explores the search space more efficiently (Bergstra and Bengio, 2012). Advanced methods like Bayesian optimization (Snoek et al., 2012) and bandit search-based approaches (Li et al., 2017) can be used as well if applicable (Bischl et al., 2023). To avoid unnecessary guesswork, the following information is expected: Hyperparameters that were searched per model (including options and ranges), the final hyperparameter settings used, number of trials, and settings of the search procedure if applicable. As tuning of hyperparameters is typically performed using specific parts of the dataset, it is essential to note that any modeling decisions based on them automatically invalidate their use as *test* data.

Models. Contemporary models (e.g. Vaswani et al., 2017; Devlin et al., 2019; Dosovitskiy et al., 2021; Chen et al., 2021; Touvron et al., 2023a,b; AI@Meta, 2024; Jiang et al., 2023a; Groeneveld et al., 2024) have very large computational and memory footprints. To avoid retraining models, and more importantly, to allow for replicability, it is recommended to save and share model weights. This may face similar challenges as those of datasets (namely, large file sizes), but it remains an impactful consideration. In most cases, simply sharing the best or most interesting model could suffice, although sharing multiple models enables more robust significance testing and allows for modeling of uncertainty through ensembling (Section 2.2.2). It should be emphasized that distributing model weights should always complement a well-documented repository as libraries and hosting sites might not be supported in the future.

Model Evaluation. The exact model and task evaluation procedure can differ significantly (e.g. Post, 2018). It is important to either reference the exact evaluation script used (including parameters, citation, and version, if applicable) or include the evaluation script in the codebase. Moreover, to ease error or post-hoc analyses, we highly recommend saving model predictions whenever possible and making them available at publication (Card et al., 2020; Gehrmann et al., 2022) and using standardized and tested implementations (e.g. Von Werra et al., 2022). Using single metrics

[37] Whether such uncertainty would be aleatoric or epistemic is difficult to decide; while more data could compensate for suboptimal hyperparameter values, it is intuitive that a model will be unlikely to converge and reduce its uncertainty for e.g. adversarially chosen values. This reinforces the argument by Baan et al. (2023) that data and model uncertainty should not be seen as a dichotomy, but rather as a spectrum.

can also distort results or paint a restrictive picture, which is why using multiple different evaluation metrics is commendable (Marie et al., 2021).

Model Cards. Apart from quantitative evaluation and optimal hyperparameters, Mitchell et al. (2019) propose model cards: A type of standardized documentation, as a step towards responsible ML and AI technology, accompanying trained ML models that provide benchmarked evaluation in a variety of conditions, across different cultural, demographic, or phenotypic and intersectional groups that are relevant to the intended application domains. They can be reported in the paper or project, and can help to collect important information for reproducibility, such as preprocessing and evaluation results. We refer to Mitchell et al. (2019); Menon et al. (2020) for examples of model cards.

Best Practices: **Codebase & Models**

◇ Publish a code repository with documentation and license;

◇ Report all details about hyperparameter search and model training;

◇ Specify the hyperparameters for replicability;

◇ Publish model predictions and evaluation scripts.;

◇ Use multiple, complementary evaluation metrics;

⋆ Use model cards;

⋆ Publish models;

3.1.3 Experiments & Analysis

Experiments and their analyses constitute the core of most scientific works, and empirical evidence is valued especially highly in ML research (Birhane et al., 2022). However, there are common issues that practitioners are faced with model training and experimental analyses, for which we discuss counter-strategies here.

Model Training. For model training, it is advisable to set a random seed for replicability, and train multiple initializations per model in order to obtain a sufficient sample size for later statistical tests. The number of runs should be adapted based on the observed variance: Using for instance bootstrap power analysis, existing model scores are raised by a constant compared to the original sample using a significance test in a bootstrapping procedure (Yuan and Hayashi, 2003; Tufféry, 2011; Henderson et al., 2018). If the percentage of significant results is low, we should collect more scores.[38] Bouthillier et al. (2021) further recommend to vary as many sources of randomness in the training procedure as possible (i.e., data shuffling, data splits etc.) to obtain a closer approximation of the true model performance. When training more runs is not feasible such as in the case of LLMs, we can for instance obtain additional observations by varying the generation process (see for instance the case study in Section 3.2.3). Nevertheless, any drawn conclusion are still surrounded by a degree of statistical uncertainty, which can be combated by the use of statistical hypothesis testing.

Significance Testing. Using deep neural networks, a number of (stochastic) factors such as the random seed (Dror et al., 2019) or even the choice of hardware (Yang et al., 2018) or framework (Leventi-Peetz and Östreich, 2022) can influence performance and need to be taken into account. First of all, the size of the dataset should support sufficiently powered statistical analyses (see Section 3.1.1). Secondly, an appropriate significance test should be chosen. We give a few rules of thumb based on Dror et al. (2018): When the distribution of scores is known, for instance a normal distribution for the Student's-t test, a *parametric* test should be chosen. Parametric tests are designed with a specific distribution for the test statistic in mind, and have strong statistical power (i.e. a lower Type II error). The underlying assumptions can sometimes be hard to verify (see Dror et al., 2018, Section 3.1), thus when in doubt *non-parametric* tests can be used. This category features tests like the bootstrap, employed in case of a small sample size, or the Wilcoxon signed-rank test (Wilcoxon, 1992), when plenty observations are available. Depending on the application, the usage of specialized tests might furthermore be desirable (Dror et al., 2019; Agarwal et al., 2021). We also want to draw attention to the fact that comparisons between multiple models and / or datasets, *require* an adjustment of the confidence level, for instance using the Bonferroni correction (Bonferroni, 1936), which is a safe and conservative choice and easily implemented for most tests (Dror et al., 2017; Ulmer et al., 2022c). Sadeqi Azer et al. (2020) provide a guide on how to adequately word insights when a statistical test was used, and Greenland et al. (2016) list common pitfalls and misinterpretations of results. Due to spatial constraints, we refer to Section 3.2 for a slightly more technical introduction to the topic. Current trends surrounding LLMs further make significance testing challenging, as training and evaluating multiple different model

[38] The resulting tensions with modern DL hardware requirements are discussed in Section 5.5.

runs can be prohibitively expensive. We explore different strategies in this restrictive setting in the case study in Section 3.2.3.

Critiques & Alternatives. Although statistical hypothesis testing is an established tool in many disciplines, its (mis-)use has received criticism for decades (Berger and Sellke, 1987; Demšar, 2008; Ziliak and McCloskey, 2008). For instance, Wasserstein et al. (2019) criticize the p-value as reinforcing publication bias through the dichotomy of "significant" and "not significant", i.e. by favoring positive results (Locascio, 2017). Instead, Wasserstein et al. (2019) propose to report it as a continuous value and with the appropriate scepticism.[39] In addition to statistical significance, another approach advocates for reporting *effect size* (Berger and Sellke, 1987; Lin et al., 2013), so for instance the mean difference, or the absolute or relative gain in performance for a model compared to a baseline. The effect size can be modeled using Bayesian analysis (Kruschke, 2013; Benavoli et al., 2017), which better fit the uncertainty surrounding experimental results, but requires the specification of a plausible statistical model producing the observations[40] and potentially the usage of markov chain Monte Carlo sampling (Brooks et al., 2011; Gelman et al., 2021). Benavoli et al. (2017) give a tutorial for applications to ML and supply an implementation of their proposed methods in a software package and guidelines for reporting details are given by Kruschke (2021), including for instance the choice of model and priors.

Best Practices: **Experiments & Analysis**

- Report mean & standard dev. over multiple runs;
- Perform significance testing or Bayesian analysis and motivate your choice of method;
- Carefully reflect on the amount of evidence regarding your initial hypotheses.

3.1.4 Discussion

Previous sections have emphasized the need to overhaul some experimental standards and have describes their interactions

[39] Or, as Wasserstein et al. (2019) note: "*statistically significant*—don't say it and don't use it".

[40] Here, we are *not* referring to a neural network, but instead to a process generating experimental observations, specifying a prior and likelihood for model scores. Conclusions are drawn from the posterior distribution over parameters of interest (e.g. the mean performance), as demonstrated by Benavoli et al. (2017).

with reducing and modeling uncertainty. But specifically with regard to statistical significance in Section 3.1.3, there is a stark conflict between the hardware requirements of modern methods (Sevilla et al., 2022) and the computational budget of the average researcher. Only the best-funded research labs can afford the increasing computational costs to account for the statistical uncertainty of results and to reproduce prior works (Hooker, 2021). Under these circumstances, it becomes difficult to judge whether the results obtained via larger models and datasets *actually* constitute substantial progress or just statistical flukes. While we present some alternatives in Section 3.2.3, this environment also make the use of traditional Bayesian DL techniques like in Section 2.2.2 more challenging. For this reason, researchers should embrace data variability as a new avenues for modeling and reducing uncertainty in large contemporary models (as discussed in Section 3.1.1).

Echoing our fundamental deliberations about the scientific process in Section 3.1, being able to (re-)produce empirical findings is critical for scientific progress, particularly in fast-growing fields like NLP (Manning, 2015). To reduce the risks of a reproducibility crisis and unreliable research findings (Ioannidis, 2005), experimental rigor is imperative. Being aware of possible harmful implications and to avoid them is therefore important, since every step can carry possible biases (Hovy and Prabhumoye, 2021; Waseem et al., 2021). This chapter aims at providing a toolbox of actionable recommendations, and a reflection and summary of the ongoing broader discussion. To improve the experimental standard in the field overall, we can distill the following suggestions: **As researchers**, we can start implementing the recommendations in this work in order to drive bottom-up change and reach a critical mass (Centola et al., 2018). **As reviewers**, we can shift focus from results to more rigorous methodologies (Rogers and Augenstein, 2021), and allow more critiques and reproductions of past works and meta-reviews to be published (Birhane et al., 2022; Lampinen et al., 2021). **As a community**, we can change the incentives around research and experiment with new initiatives. With concrete best practices to raise awareness and a call for uptake, we hope to aid researchers in their empirical endeavors. The rest of this chapter is dedicated to the practice of statistical hypothesis testing and its challenges in the era of LLMs.

3.2 Statistical Hypothesis Testing

The following work is based on Ulmer et al. (2022c).

In this part of the chapter, we are discussing statistical hypothesis testing with an application to comparing two models or *algorithms*. While terms like model or algorithms will be used almost synonymously in the rest of this thesis, it will aid the rest of this chapter to define these notions better.

Definition 3 (Model)**.** We define a model $f_{\boldsymbol{\theta}}$ to be the element of some hypothesis class $f_{\boldsymbol{\theta}} \in \mathbb{H}$. Here, the hypothesis class is loosely defined as all neural predictors trained using the same architecture and training data.

Importantly, the above definition does not imply that all predictors $\mathbb{H}$ comprise the same parameter values—they can be influenced by factors such as random seeds or the order of training samples, and in the case of LLMs, the use of different generation hyperparameters or prompt templates.

Definition 4 (Metric & Observation)**.** Let us define $\phi : \mathbb{H} \times \mathcal{P}(\mathbb{D}) \to \mathbb{R}$ to be a function measuring the performance of a predictor f_{θ} on some dataset $\mathbb{D} \in \mathcal{P}(\mathbb{D})$ in form of a real number $s \in \mathbb{R}$, called *observation* or *score*, with ϕ called the *metric*.

We will assume in the following that a higher number for s indicates a more desirable behavior. Now, we let $\mathbb{S}_{\mathbb{A}}$ denote a set of observations obtained from different instances of a specific hypothesis class $\mathbb{A}$. Ideally for deep neural networks, obtaining a set of observations $\mathbb{S}_{\mathbb{A}}$ would involve training multiple *instances* of a network with the same architecture using different sets of hyperparameters and random initializations. Since the former part often becomes computationally infeasible in practice, we follow the advice of Bouthillier et al. (2021) and assume that it is obtained by fixing one set of hyperparameters after a prior search and varying as many other random elements as possible. Here, we only give a very brief introduction into statistical hypothesis testing using p-values, and refer the reader to resources such as Japkowicz and Shah (2011); Dror et al. (2018); Raschka (2018); Sadeqi Azer et al. (2020); Dror et al. (2020); Riezler and Hagmann (2021) for a more comprehensive overview. Using the introduced notation, we can define a one-sided test statistic $\delta(\mathbb{S}_{\mathbb{A}}, \mathbb{S}_{\mathbb{B}})$ based on the gathered observations. An example of such test statistics is for instance the difference in observation means $\delta(\mathbb{S}_{\mathbb{A}}, \mathbb{S}_{\mathbb{B}}) = \hat{\mu}_{\mathbb{A}} - \hat{\mu}_{\mathbb{B}}$ with $\mu_{(\cdot)} = \frac{1}{|\mathbb{S}_{(\cdot)}|} \sum_{s_i \in \mathbb{S}_{(\cdot)}} s_i$. We then formulate the following null hypothesis:

$$\mathrm{H}_0: \ \delta(\mathbb{S}_{\mathbb{A}}, \mathbb{S}_{\mathbb{B}}) \leq 0. \tag{3.1}$$

The null hypothesis H_0 assumes the opposite of our desired case, namely that $\mathbb{A}$ is not better than $\mathbb{B}$, but equally as good or worse, as indicated by the value of the test statistic. Usually, the goal becomes to reject this null hypothesis. p-value testing is a frequentist method in the realm of statistical hypothesis tests. It introduces the notion of data that *could have been observed* if we were to repeat our experiment again using the same conditions, which we will write with superscript $^{\text{rep}}$ in order to distinguish them from our actually observed scores (Gelman et al., 2021). We then define the p-value as the probability that, under the null hypothesis H_0, the test statistic using replicated observations is larger than or equal to the *observed* test statistic:

$$p(\delta(\mathbb{S}_{\mathbb{A}}^{\text{rep}}, \mathbb{S}_{\mathbb{B}}^{\text{rep}}) \geq \delta(\mathbb{S}_{\mathbb{A}}, \mathbb{S}_{\mathbb{B}}) \mid \mathrm{H}_0). \tag{3.2}$$

We can interpret this expression as follows: Assuming that $\mathbb{A}$ is not better than $\mathbb{B}$, the test assumes a corresponding distribution of statistics that δ is drawn from. So how does the observed test statistic $\delta(\mathbb{S}_{\mathbb{A}}, \mathbb{S}_{\mathbb{B}})$ fit in here? This is what the p-value expresses: When the probability is high, $\delta(\mathbb{S}_{\mathbb{A}}, \mathbb{S}_{\mathbb{B}})$ is in line with what we expected under the null hypothesis, so we *cannot* reject the null hypothesis, or in other words, we *cannot* conclude $\mathbb{A}$ to be better than $\mathbb{B}$. If the probability is low, that means that the observed $\delta(\mathbb{S}_{\mathbb{A}}, \mathbb{S}_{\mathbb{B}})$ is quite unlikely under the null hypothesis and that the reverse case is more likely—i.e. that it is likely larger—and we conclude that $\mathbb{A}$ is indeed better than $\mathbb{B}$. In summary, the question that a p-value asks can be stated as follows: Assuming the null hypothesis to be true, how likely is a test statistic to be at least as extreme as observed? Note that **the p-value does not express whether the null hypothesis is true**. To make our decision about whether or not to reject the null hypothesis, we typically determine a threshold—the significance level α, often set to 0.05—that the p-value has to fall below. However, it has been argued that a better practice involves reporting the p-value alongside the results without a pigeonholing of results into significant and non-significant (Wasserstein et al., 2019).

3.2.1 Almost Stochastic Order

Deep neural networks are known to be highly non-linear models (Li et al., 2018), having their performance depend to a large extent on the choice of hyperparameters, random seeds and other (stochastic) factors (Bouthillier et al., 2021). This makes comparisons between algorithms more difficult, as illustrated by the motivating example below by Dror et al. (2019):

Example 1 (Part-of-Speech tagging)**.** *Consider the results for Part-of-Seech-tagging given in the table on the right, taken over 3898 and 1822 observations using different hyperparameter configurations and random seeds, respectively. Optimizing with Adam (Kingma and Ba, 2015) gives a higher average word-level accuracy than using RMSprop (Tieleman and Hinton, 2012), however the median favors the latter. Furthermore, the minimum across a few runs favor Adam, but the maximum is higher for RMSprop. So, which algorithm do we consider to be* better*?*

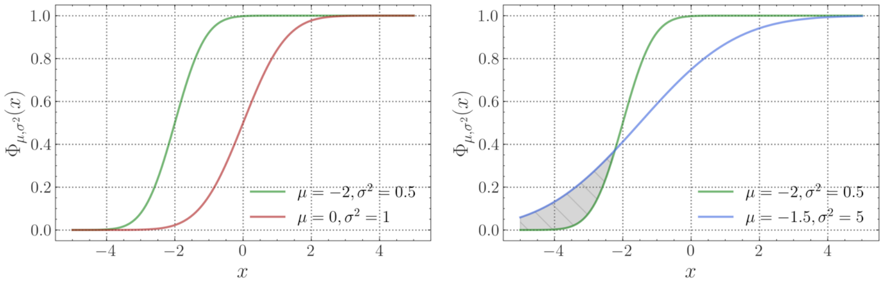
	Adam	RMSprop
Mean	.9224	.9190
Std. dev.	.0604	.0920
Median	.9319	.9349
Min.	.1746	.1420
Max.	.9556	.9573

Therefore, Dror et al. (2019) propose *almost stochastic order* (ASO) for Deep Learning models based on the work by del Barrio et al. (2018a).[41] It is based on a relaxation of the concept of *stochastic order* by Lehmann (1955): A random variable $x_{\mathbb{A}}$ is defined to be *stochastically larger* than $x_{\mathbb{B}}$ (denoted $x_{\mathbb{A}} \succeq x_{\mathbb{B}}$) if $\forall x : F(x) \leq G(x)$, where F and G denote the cumulative distribution functions (CDF) of the two random variables. The CDF is defined as $F(t) = p(x \leq t)$, while the *empirical* CDF given a sample $\{x_1, \ldots, x_n\}$ is defined as

$$F_n(t) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\big(x_i \leq t\big),$$

with $\mathbb{1}(\cdot)$ being the indicator function. In practice, since we do not know the real score distributions $p(x_{\mathbb{A}})$ and $p(x_{\mathbb{B}})$, we cannot use the precise CDFs in subsequent calculations, and we rely on the empirical CDFs F_N and G_M. A case of stochastic order is illustrated in Figure 3.3a, using the CDFs of two normal distributions. However, in cases such Figure 3.3b we would still like to declare one of the algorithms superior, even though the stochastic order of the underlying CDFs is partially violated. Several ways to quantify the violation of stochastic dominance exist (Álvarez-Esteban et al., 2017; del Barrio et al., 2018b), but here we elaborate on the optimal transport approach by del Barrio et al. (2018a). They propose a the following expression quantifying the distance of each random variables from being stochastically larger than the other:

[41] Implementation details and pseudo-code are given in Appendix C.3.



(a) Stochastic order with red $\succeq$ green. (b) Almost stochastic order, with blue $\succsim$ green.

Figure 3.3: Examples for stochastic order (a) and almost stochastic order (b), illustrated using the CDFs of two normal random variables. Because stochastic order is too strict to be practical, almost stochastic order allows for some degree of violation of the order (gray area in (b)).

$$\varepsilon_{W_2}(F, G) = \frac{\int_{\mathbb{V}_x} (F^{-1}(t) - G^{-1}(t))^2 dt}{(W_2(F, G))^2}, \tag{3.3}$$

with the *violation ratio* $\varepsilon_{W_2}(F, G) \in [0, 1]$ and a *violation set* $\mathbb{V}_x = \big\{t \in (0, 1) : F^{-1}(t) < G^{-1}(t)\big\}$, i.e. where the stochastic order is being violated. Equation (3.3) contains the following components: Firstly, the quantile functions $F^{-1}(t)$ and $G^{-1}(t)$ associated with the corresponding CDFs:

$$F^{-1}(t) = \inf\big\{x : t \leq F(x)\big\}, \quad t \in (0, 1).$$

The quantile functions allow us to define stochastic order via $X \succeq Y \iff \forall t \in (0, 1) : F^{-1}(t) \geq G^{-1}(t)$. Secondly, it comprises the univariate l_2-Wasserstein distance:

$$W_2(F, G) = \sqrt{\int_0^1 \big(F^{-1}(t) - G^{-1}(t)\big)^2 dt}, \tag{3.4}$$

which for univariate functions can be expressed through their inverse CDFs (De Angelis and Gray, 2021). Finally, del Barrio et al. (2018a); Dror et al. (2019) define a hypothesis test based on this quantity by formulating the following hypotheses:

$$\begin{aligned} \mathrm{H}_0 &: \varepsilon_{W_2}(F, G) \geq \tau \\ \mathrm{H}_1 &: \varepsilon_{W_2}(F, G) < \tau, \end{aligned}$$

for a pre-defined threshold $\tau > 0$, for instance 0.5 or lower (see discussion in Appendix B.1 about the choice of threshold). Further, Álvarez-Esteban et al. (2017); Dror et al. (2019) produce a frequentist upper bound to this quantity, defining the minimal ε_{W_2} for which we can reject the null hypothesis with a confidence of $1 - \alpha$ as

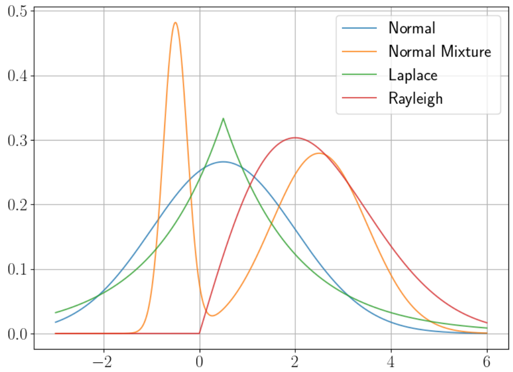


Figure 3.4: Plot of distributions used to empirically test the Type I and Type II error of significance tests in Section 3.2.2.

$$\varepsilon_{\min}(F_N, G_M, \alpha) = \varepsilon_{W_2}(F_N, G_M) - \sqrt{\frac{N+M}{NM}}\hat{\sigma}_{N,M}\Phi^{-1}(\alpha). \quad (3.5)$$

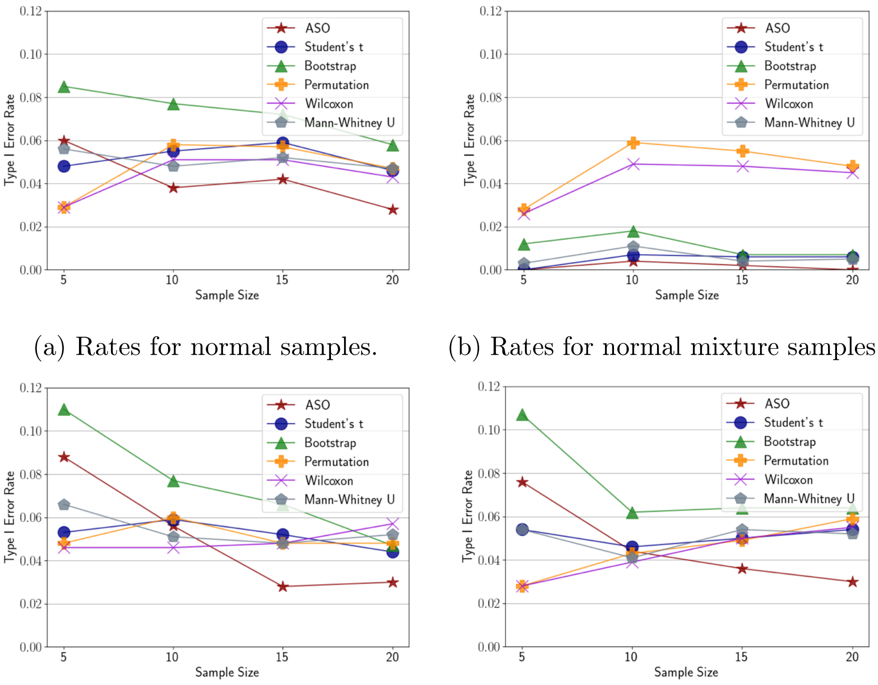
The variance term $\hat{\sigma}_{N,M}$ is estimated using a bootstrapping estimator (as introduced in Section 2.1.1) for the variance, with F_N^* and G_M^* denoting empirical CDFs based on sets of scores resampled from original sets of model scores, similar to re-sampling procedure in other tests like the bootstrap (Efron and Tibshirani, 1994) or permutation-randomization test (Noreen, 1989):

$$\hat{\sigma}_{N,M}^2 = \text{Var}\left[\sqrt{\frac{NM}{N+M}}\big(\varepsilon_{W_2}(F_N^*, G_M^*) - \varepsilon_{W_2}(F_N, G_M)\big)\right]. \quad (3.6)$$

Thus, if $\varepsilon_{\min}(F_N, G_M, \alpha) < \tau$, we can reject the null hypothesis and claim that algorithm $\mathbb{A}$ is better than $\mathbb{B}$, with a growing discrepancy in performance the smaller the value becomes.

3.2.2 Experimental Comparison

We compare ASO to established significance tests such as the Student's-t, the bootstrap (Efron and Tibshirani, 1994), and the permutation-randomization test (Noreen, 1989), along with the Wilcoxon signed-rank (Wilcoxon, 1992) and Mann-Whitney U test (Mann and Whitney, 1947) on different types of distributions, which are plotted in Figure 3.4. We plot the Type I error rate per 500 simulations for ASO and 1000 simulations for the other tests as a function of sample size in Figure 3.5, where we sample both sets of observation from the same distribution. For Figure 3.5a, we sample from $\mathcal{N}(0, 1.5^2)$ and try a bimodal normal mixture in



(a) Rates for normal samples.

(b) Rates for normal mixture samples.

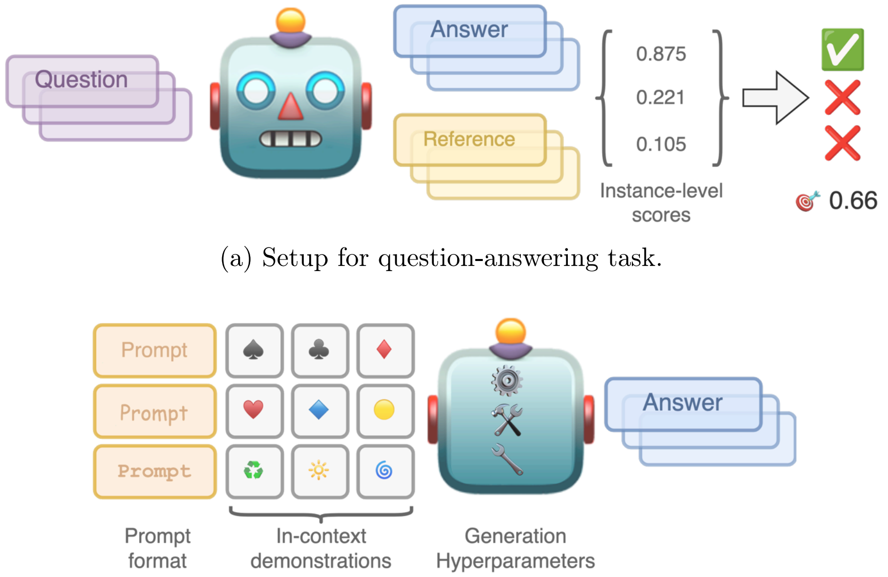
(c) Rates for Laplace samples.

(d) Rates for Rayleigh samples.

Figure 3.5: Comparing type I error rates for different tests and distributions as a function of sample size. Decisions are made using a confidence threshold of $\alpha = 0.05$ and $\tau = 0.2$ for ε_{min}.

Figure 3.5b (using the same parameter for the second component, and $\mathcal{N}(-0.5, 0.25^2)$ with mixture weights $\pi_1 = 0.75$ and $\pi_2 = 0.25$). To test the behavior of tests on non-normal distributions, we also sample from a Laplace$(0, 1.5^2)$ distribution in Figure 3.5c, which possesses a different behavior around the main, as well as the Rayleigh distribution with Rayleigh(1) in Figure 3.5d, which has a heavy tail. We can see that ASO performs either en par or better than other tests in all scenarios, achieving *lower* error rates the more samples are available, while other tests score around the expected type I error of 5%. In Appendix B.1, Type II error experiments reveal that the test produces comparatively higher error rates for ASO, though. This can be explained by the fact that we use the upper bound ε_{min} instead of ε_{W_2} to evaluate the null hypothesis, which makes the test act more conservatively. We also find in Appendix B.1 that a decision threshold of $\tau = 0.2$ strikes an acceptable balance between Type I and II error rates across different scenarios. Overall, we argue that a lower Type I error is more advantageous in the context of empirical research, and that a *decreasing* error rate w.r.t. higher sample sizes constitutes an appealing property when used on arbitrary distributions. In these experiments, the score distributions were determined *a priori* in order to create rigid experimental conditions. Naturally, a practitioner would not know these distribution in a typical setting, which is why we illustrate the usage of the test in the next section.

3.2.3 Case study: Question-Answering with Large Language Models



(a) Setup for question-answering task.

(b) Strategies to produce varying answers.

Figure 3.6: Setup for the question-answering case study. In (a), we depict the general task setup: Questions are given to an LLM, which produces answers that are scored against reference answers using ROUGE-L. The scores for every question-answer pair are compared against a pre-defined threshold, which determines whether an answer is considered correct, and an accuracy score can be computed. (b) In order to produce different answers for the same question, we can vary different factors, including the prompt format, the in-context demonstrations, generation hyperparameters, or all of these factors together.

We apply the ASO test to a very relevant problem in NLP: Comparing the results from different LLMs, where models are already trained and multiple different seeds are not available. Here, we explore ways in which we can still enable statistical hypothesis testing despite the more restrictive setup. While we do not quantify any uncertainty in model *predictions* here, introducing variability and employing hypothesis testing enables us to quantify uncertainty in model *results*, therefore aiding model selection.

Setup. We use three popular open-source models, namely MosaicAI MPT 7B (MosaicML NLP Team, 2023), Mistral 7B

(Jiang et al., 2023a), and OLMo 7B (Groeneveld et al., 2024),[42] and compare them on a closed-book question-answering task on TriviaQA (Joshi et al., 2017). The general task setup is shown in Figure 3.6a: Given a number of questions from the TriviaQA test set, we obtain the LLM's answers, which are scored against reference answers using ROUGE-L (Lin, 2004), which is a measure based on n-gram overlap. If the obtained score surpasses a pre-defined threshold, we score an answer as correct. From this, we obtain a single accuracy score for the whole test set. In each case, we use their default generation methods set for the model on the HuggingfaceHub and 10 other instances as in-context examples.

The goal is to show that even when we operate with monolithic models, we can still facilitate meaningful comparisons using statistical hypothesis testing. The default option usually consist of just comparing the two accuracies (scalar comparison), however this does not take any uncertainty in the results into account. Instead, we might compare the population of instance-level scores in Figure 3.6a before thresholding (instance-level comparison), or use a bootstrap estimator on the instance-level scores to obtain multiple accuracy scores, similar to our estimation of the probability of heads using a sample of bootstrapped coin flips in Section 2.1.1 (bootstrapping comparison). Another approach is to vary the factors that produce an LLM's answer, which are depicted in Figure 3.6b: We can for instance change the prompt formatting (multi-prompt comparison), change the in-context demonstrations by re-sampling them from the training set for each inference (varying in-context samples), or modify the hyperparameters that influence the models generation (generation hyperparameters). Lastly, we can also combine prompt formatting, varying in-context examples and generation hyperparameters by changing them jointly for every test instance (mix). We briefly discuss each of these options in more detail.

Scalar Comparison. We first consider the potentially most common form of comparison, namely single scalars. For this purpose, we compute the accuracy per model on the given test set of questions. To judge whether a question has been answered correctly, we use the same heuristic as employed by Kuhn et al. (2023), where we compute the ROUGE-L score (Lin, 2004) as implemented by the `evaluate` package[43] between a given model answer and

[42] More precisely, we use `mosaicml/mpt-7b`, `mistral-community/Mistral-7B-v0.2`, and `allenai/OLMo-1.7-7B-hf`.

[43] See https://huggingface.co/docs/evaluate/index.

gold answer. When the resulting score surpasses a value of 0.3, an answer is considered correct.

Instance-level Comparison. Instead of aggregating the measurements on all test instances into a single score, we can instead look at them as a set of observations. This enables us to compare larger populations of observations, as opposed to having only one single observation per model. For question-answering, we use the ROUGE-L scores, but without applying a threshold. A key difference to the other tested approaches is that this comparison answers a subtly different question about the models: Instead of considering which hypothesis class of model is better by evaluating different model instances after training them with distinct random seeds, we instead ask which trained model *instance* tends to give better-scored answers in general (as judged by the ROUGE-L heuristic).

Bootstrapping Comparison. In Section 2.1.1, we discussed bootstrapping as a way to quantify the uncertainty about a quantity of interest. We can apply the same technique to the accuracy by bootstrapping samples of observations from the existing set of answered questions, and computing the accuracy on these pseudo-samples. These scores can then be used to compute the standard error and to run them through the ASO test.

Multi-prompt Comparison. LLMs can be very sensitive to the chosen prompt format (Mizrahi et al., 2024; Sclar et al., 2023). Therefore, instead of evaluating predictions from models trained with different random seeds, we can instead consider predictions from the same model but *using different prompts*, and treat the resulting accuracies as observations. Specifically, we test the following prompt templates:

1. `{question}`, `Q: {question} A:`
2. `Question: {question} Answer:`
3. `Take the following question: '{question}'. Give the correct answer:`

Varying In-context Examples. Various studies have pointed out the importance of in-context samples for task-specific model capabilities (Xie et al., 2022; Min et al., 2022; Hendel et al., 2023). For this reason, we run four additional evaluations where we randomly sample a new set of in-context samples.

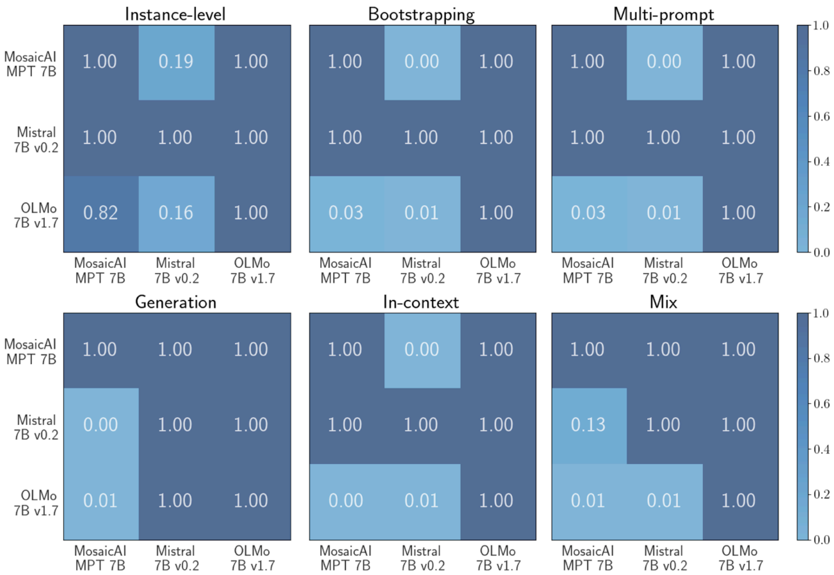
Generation Hyperparameters. We also consider generating answers using different generation parameters. Specifically, we try the default approach for the three models, greedy decoding, as well as Nucleus sampling (Holtzman et al., 2020) with $p = 0.9$, top-k sampling (Fan et al., 2018; Holtzman et al., 2018; Radford et al., 2019) with $k = 60$ or beam search with three beams. Lastly, we also combine this with multiple different prompts and different in-context samples, where we answer every question 5 times, each time sampling a different prompt, generation configuration, and in-context demonstrations randomly.

Results. All accuracies for the different methods including standard deviations are shown in Figure 3.7a, with an overview of all the ε_{min} values calculated by the ASO test in Figure 3.7b. Recall that according to Section 3.2.1, we would declare one model superior to another when $\varepsilon_{\text{min}} < \tau$, which empirically $\tau = 0.2$ to provide a good trade-off between Type I and Type II error. We can see that the ordering of models largely agrees across settings, but can provide subtle differences. All models usually generate through greedy sampling. When using different generation hyperparameters, we can observe a noticeable degradation in results, although the OLMo model seems to be most robust to changes in generation parameters. Interestingly, the ε_{min} values in Figure 3.7b show that all evaluations mostly agree in their result; however the comparison of instance-level scores seems to underestimate the degree of almost stochastic dominance (as shown through larger ε_{min} values for the best models). A noticeable exception for this agreement is the experiment using different generation hyperparameters, where the severe loss in performance renders none of the results significant. In the end, the mixture of a random prompt template, generation hyperparameters and in-context examples seems to portray the clearest picture of the model rankings through the ε_{min} values.

Formalization. In this case study, we discussed a number of ways we can use to perform statistical hypothesis testing using LLMs, assuming access to an already trained model. All of these are subtly different in what the kinds of uncertainties they take into account to compare models. To investigate the differences, we formalize the problem: Let $\mathbf{x}$ be shorthand for an input sequence, $\mathbf{y}$ for a generated sequence and ϕ a function mapping a generated sequence to an evaluation score (e.g. an indicator function deciding whether an answer is correct). Further, let $\boldsymbol{\rho}$ be a prompt template, $\boldsymbol{\gamma}$ a set of generation parameters, $\mathcal{C}$ a set of in-context demonstrations and $\boldsymbol{\lambda}$ a set of training hyperparameters (including architecture, optimizer, regularization, finetuning strategy etc.).

	Accuracy					
Model	Scalar	Bootstrapping	Multi-prompt	Generation	In-context	Mix
MosaicAI MPT 7B	.49	.49 ± .00	.51 ± .01	.02 ± .02	.51 ± .01	.30 ± .02
Mistral 7B v0.2	.37	.37 ± .00	.40 ± .04	.15 ± .09	.43 ± .04	.33 ± .03
OLMo 7B v1.7	**.51**	**.51 ± .01**	**.57 ± .04**	**.17 ± .02**	**.59 ± .03**	**.40 ± .03**

(a) Evaluation results, given in accuracy. Best results are bolded, significant differences according to the ASO test are underlined. Shown are results from a scalar comparison (Scalar), bootstrapping instance-level observations (Bootstrapping), trying different prompt templates (Multi-prompt), generation hyperparameters (Generation), in-context demonstrations (In-context), or randomly sampling a prompt, generation settings and in-context examples for each input (Mix). Instance-level results were only used to perform hypothesis testing for the scalar results, and are therefore not included as a column.



(b) $\varepsilon_{\min}$ values comparing the LLMs using different sets of observations.

Figure 3.7: Results for the case study. Given are (a) accuracy scores, either as single scalar or accuracy scores with confidence intervals as a result of bootstrapping or using multiple-prompts. Further shown are (b) the $\varepsilon_{\min}$ scores based on the instance-level observations, bootstrapping observations as well using multiple prompts, different generation parameters, different in-context demonstrations or a combination of the last tree (Mix).

We are then interested in two quantities: Aggregate metrics such as accuracy, which we can formulate as the expected value of ϕ under the model on a given dataset, and the expected accuracy arising when varying all the other factors mentioned above, forming another expectation:

$$\mathbb{E}_{p(\boldsymbol{\gamma},\boldsymbol{\rho},\mathcal{C})}\Big[\mathbb{E}_{p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta},\boldsymbol{\rho},\boldsymbol{\gamma})}\big[\phi(\mathbf{y})\big]\Big] = \int\dots\int \underbrace{\phi(\mathbf{y})}_{\text{Score}}\underbrace{p(\mathbf{y} \mid \boldsymbol{\theta},\mathbf{x},\boldsymbol{\rho},\gamma,\mathcal{C})}_{\text{LLM Predictive Dist.}}\ \underbrace{p(\boldsymbol{\rho})p(\gamma)p(\mathcal{C})}_{\text{Generation Priors}} \\ p(\boldsymbol{\theta} \mid \mathbb{D},\boldsymbol{\lambda})p(\boldsymbol{\lambda})\mathrm{d}\mathbf{y}\,\mathrm{d}\boldsymbol{\theta}\,\mathrm{d}\gamma\,\mathrm{d}\boldsymbol{\rho}\,\mathrm{d}\mathcal{C}\mathrm{d}\boldsymbol{\lambda}\,. \tag{3.7}$$

We can use this to analyze all the test setups above by applying Dirac delta functions (as previously used in Equation (2.33)) and Monte Carlo integration (see Equation (2.49)) to evaluate Equation (3.7). For instance, the scalar comparison assume an single prompt $\hat{\boldsymbol{\rho}}$, set of generation parameters $\hat{\boldsymbol{\gamma}}$, in-context samples $\hat{\mathcal{C}}$ and weights $\hat{\boldsymbol{\theta}}$ and thus Equation (3.7) becomes

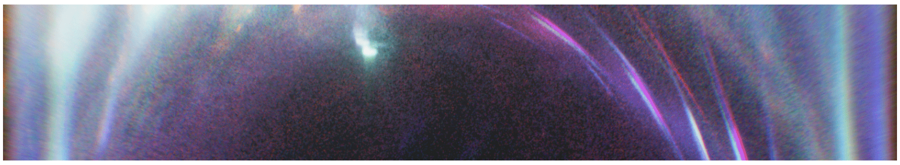
$$\begin{aligned}
&\mathbb{E}_{p(\boldsymbol{\gamma},\boldsymbol{\rho},\mathcal{C})}\Big[\mathbb{E}_{p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta},\boldsymbol{\rho},\boldsymbol{\gamma})}\big[\phi(\mathbf{y})\big]\Big] \\
&\approx \int \phi(\mathbf{y})p(\mathbf{y} \mid \boldsymbol{\theta},\mathbf{x},\boldsymbol{\rho},\boldsymbol{\gamma},\mathcal{C})\delta(\boldsymbol{\rho}-\hat{\boldsymbol{\rho}})\delta(\boldsymbol{\gamma}-\hat{\boldsymbol{\gamma}})\delta(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})\delta(\mathcal{C}-\hat{\mathcal{C}})\mathrm{d}\mathbf{y} \\
&= \int \phi(\mathbf{y})p(\mathbf{y} \mid \hat{\boldsymbol{\theta}},\mathbf{x},\hat{\boldsymbol{\rho}},\hat{\boldsymbol{\gamma}},\hat{\mathcal{C}})\mathrm{d}\mathbf{y}.
\end{aligned} \tag{3.8}$$

The same assumptions are also applied for the instance-level comparison, with the difference that we only evaluate the outer expectation in Equation (3.7). Further, we can interpret the bootstrapping procedure as a different outer expectation in Equation (3.7), where we instead evaluate the expectation over all possible samples (with replacement) of our original set of generated sequences. The conclusion we can draw from this is the following: To evaluate the overall performance of a model, we would like to approximate Equation (3.7) as closely as possible, ideally by performing a full ancestral sampling scheme. For LLMs, this is not feasible, since we often have to work with a single, already trained model. Bouthillier et al. (2021) have unveiled the perhaps counter-intuitive intuition that *increasing* amount of randomness in our experiments actually helps to *decrease* the variance of our estimate of Equation (3.7). We follow this idea and vary as many aspects as possible, which in this case study produces a clear ranking of the robustness of a model. For language models, this implies running the model over the dataset multiple times, but sampling different generation parameters and prompt templates like in our mix variant (as advocated for by Mizrahi et al., 2024). In cases where running the model multiple times for each input might still be prohibitively expensive, we can always fall back onto a bootstrap estimator.

3.2.4 Discussion

The previous sections have demonstrated the advantages of the ASO test in an neural network setting. Nevertheless, using these techniques in practice comes with limitations as well, which the end user should be aware of. The first line of limits comes with ASO itself. Multiple steps of the procedure require different kinds of approximations or properties that are only guaranteed to hold in the infinite-sample limit, e.g. the bootstrap estimator of the variance in Equation (3.6). Furthermore, significance tests in general are known to sometimes provide unreliable results with small (Reimers and Gurevych, 2018) or very large sample sizes (Lin et al., 2013), are prone to misinterpretation (Gibson, 2021; Greenland et al., 2016), and encourages binary significant / non-significant thinking (Wasserstein et al., 2019; Sadeqi Azer et al., 2020). Bayesian analysis (Kruschke, 2013; Benavoli et al., 2017; Gelman et al., 2021) is therefore an attractive alternative to statistical hypothesis testing, where the user draws conclusions from posterior distributions over quantities of interest. A potential drawback of this methodology is that it often comes at the cost of having to use Markov chain Monte Carlo methods, which require experience from the user to validate convergence and defining appropriate models and model priors.

For the application to LLMs, Section 3.2.3 has demonstrated that even with a fully trained model, we can still perform meaningful statistical hypothesis testing by either using bootstrapping or by varying prompt templates, generation hyperparameters and in-context demonstrations. Some of these methods for model comparison will now be used in the remaining chapters of this thesis.



4 Uncertainty in Text Classification

“*Two roads diverged in a yellow wood,*
And sorry I could not travel both
And be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth;

Then took the other, as just as fair,
And having perhaps the better claim,
Because it was grassy and wanted wear;
Though as for that the passing there
Had worn them really about the same,

And both that morning equally lay
In leaves no step had trodden black.
Oh, I kept the first for another day!
Yet knowing how way leads on to way,
I doubted if I should ever come back.

I shall be telling this with a sigh
Somewhere ages and ages hence:
Two roads diverged in a wood, and I—
I took the one less traveled by,
And that has made all the difference.”

—*The Road Not Taken* by Robert Frost (1915).

Assume we would like to automate the moderation of postings on a social media platform. While it would be preferable to always use human moderators, this is often not feasible due to the deluge of posts, and also not desirable due to the psychological impact that the moderation of harmful content can have. After having trained a classifier on some labeled training instances, we are ready to deploy. And while we expect a large number of the flagged cases to be clear positives, there will inadvertently be instances for which the classifier struggles, for example sentences in which an toxic remark is quoted or lacks context. The sentence below was taken from the Wikitalk dataset (Wulczyn et al., 2017; Borkan et al., 2019), which includes discussions among Wikipedia editors:

> “*I was responding to a post by AndyTheGrump at Talk: Communist terrorism, section ‘Marxism is not the only ‘communism”, where he called me ‘idiot’ and then refused to retract his remark when I requested him to do so.*”

The mention of “idiot” here might already set off the toxicity classifier, even though the sentence just quotes another user’s remark. In this case, we might want to defer to the decision to a human moderator when the classifier shows uncertainty. To make the task of moderation easier, we could also employ another system to label the spans of text that contain harmful speech. Here, we might only show the parts that the system is most uncertain about to limit the exposure to toxicity, and potentially ask the moderator to label them in order to improve the training data for future model updates. To illustrate this, let us look at another (truncated) example from the dataset, labeling it in two different ways (assuming simplified tokenization):

I'd	like	to	offer	you	a	great	big	glass	of	shut-the-f∗@#-up	juice
−	−	−	−	−	−	−	−	−	−	+	−
O	O	O	O	O	B-TOX	I-TOX	I-TOX	I-TOX	I-TOX	I-TOX	I-TOX

In the first annotation, we focus on whether single words could be considered toxic or not, while in the second annotation, we capture an entire toxic span using BIO-tags (which indicate the **b**eginning, **i**nside or **o**utside of such a phrase). What we outlined above are instances of classic NLP task formats, namely *sequence classification* and *sequence labeling*, respectively. In the former we simply assign a label to an entire sequence, whereas in the latter we label or classify parts of a sequence. In this thesis, we will refer to both jointly as *text classification*. Sequence labeling subsumes tasks such as part-of-speech tagging,[44] where the labels can e.g. be noun, verb or adverb, or *named entity recognition*, in which we identify named entities such as people, organization or locations. In this work we will use the terms *label* and *class* interchangeably to refer to a category from a set of categories that is assigned to (part of) an input.[45]

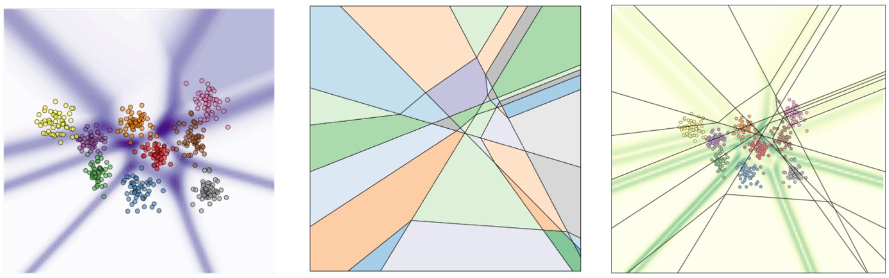
However, quantifying the uncertainty in these decisions is challenging. Uncertainty is not always present in predictions when we might expect them, and might be present if it is unwarranted. This chapter aims to understand this behavior, both from a theoretical and empirical perspective. Therefore, we demonstrate some shortcomings of UQ with ReLU networks in the next section, before returning to text classification in Section 4.2.

[44] PoS tagging also is common preprocessing step for parser that produce parse trees as the ones shown in Section 2.1.3.

[45] Even though this chapter focuses on *multi-class classification*, there is a subtle difference to *multi-label classification*: Multi-class means that we have multiple choices, but only one of them will be considered correct at a time, which makes sense when trying to choose from mutually exclusive options. In the multi-label classes, we choose for each possible label whether it is applicable or not, allowing multiple labels to be correct at the same time. For instance, when classifying legal judgments according to which human right articles are being violated, we can find that each judgement can violate multiple articles at once (Chalkidis et al., 2019b).

4.1 Theoretical Pitfalls in Classification

The following work is based on Ulmer and Cinà (2021).



(a) Predictive entropy. (b) Polytopal regions. (c) Magnitude of predictive entropy gradient.

Figure 4.1: Uncertainty and linear regions of a ReLU classifier trained on example data. (a) Uncertainty measured by predictive entropy on synthetic data, illustrated by increasing shades of purple, with white denoting absolute certainty. (b) Polytopal, linear regions in the feature space induced by the same classifier (as introduced by Arora et al., 2018, plotted using the code by Jordan et al., 2019). (c) Gradient norm of the predictive entropy plotted in shades of green—small perturbations in the input have a decreasing influence on the uncertainty of the network as we stray away from the training data, creating large areas in which uncertainty levels are overgeneralized.

It is well-known that neural network classifiers tend to be overconfident in their predictions (Guo et al., 2017; see more related work in Section 2.2.1). In addition, they can exhibit high levels of certainty when this is unwarranted, and often fail to correctly identify OOD samples (Snoek et al., 2019; Nalisnick et al., 2019b). Ulmer et al. (2020) showed that even techniques specifically developed to quantify the model's uncertainty struggle at detecting OOD samples for a relatively simple classification task. Crucially, it was shown that neural discriminators tend to project vast areas of high certainty far away from the training distribution—a behavior that seems completely at odds with reliable OOD detection. These observations are replicated in Figure 4.1: In Figure 4.1a, we can see that the predictive entropy of a ReLU classifier displays low uncertainty in large regions behind the observed data clusters. As Arora et al. (2018) showed, ReLU classifiers induce polytopal linear regions in the feature space shown in Figure 4.1b, which was used by the previous work of (Hein et al., 2019) to show that the network's confidence is an unsuitable measure of uncertainty to detect OOD inputs. However, the reasons for this behavior in a classification setting are less studied, and thus we study this behavior on additional uncertainty metrics such as predictive entropy in Figure 4.1c.

In this chapter, we present a theoretical argument to explain such phenomena, showing that certainty levels are generalized on sub-spaces defined by the network (see Figures 4.1b and 4.1c). We do this by simulating covariate shift for single feature values of real variables and studying the asymptotic behavior of the model. Our first result shows that, under mild assumptions about the network's behavior on certain subspaces, ReLU-based neural network classifiers coupled with widely used uncertainty metrics always converge to a fixed uncertainty level on OOD samples. We extend this result by proving that variational inference-based and ensembling methods in combination with several uncertainty estimation techniques suffer from the same problem (Theorem 1). This phenomenon is illustrated and discussed on synthetic data. These results entail that, when the conditions of the theorem are met, these models cannot be used to reliably detect OOD: since the level of certainty is generalized from seen to unseen data, the models are unable to differentiate between the two. The findings of this chapter have bearings on OOD detection for several critical applications using neural classifiers with ReLU activation functions, and I will also discuss the impact on the following experiments for NLP.

4.1.1 Preliminaries

We first introduce some relevant definitions for the rest of this chapter.

Out-of-distribution Data. A Although there exist many different notions of dataset shift (Shimodaira, 2000; Moreno-Torres et al., 2012; Hupkes et al., 2023), we particularly focus on *covariate shift*, in which the distribution of feature values—the covariates—differs from the original training distribution $p(\mathbf{x})$. We focus on this kind of shift as it is especially common in non-stationary environments like healthcare (Curth et al., 2019), where distributional drifts over time are very common. To simulate covariate shift, we obtain OOD samples by shifting points away from the training distribution by means of a scaling factor. This approach is in line with recent experiments on covariate shift and OOD detection (Snoek et al., 2019; Ulmer et al., 2020). We would expect a reliable OOD detection model to display increasing uncertainty as points stray further and further away from the mass of $p(\mathbf{x})$, thus we study the behavior of OOD detection models in the limit, when the scaling factor is allowed to grow indefinitely in at least one dimension.

Uncertainty Metrics. We begin by first defining a neural discriminator in the form of a ReLU classifier, which we assume to follow common architectural conventions. Thus, it consist of a series of affine transformations with ReLU (Glorot et al., 2011) activation functions, defined by $\text{ReLU}(x) = \max(0, x)$. Together with a final softmax function (Bridle, 1990) as defined in Equation (0.2), it parameterizes a categorical distribution over classes.[46]

Definition 5 (ReLU Classifier)**.** Let $\mathbf{x} \in \mathbb{R}^D$ be an input vector and K the number of classes in a classification problem. The unnormalized output of the network after L layers is a function $f_{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}^K$ with the final output following after an additional softmax function $\bar{\sigma}(\cdot)$ s.t. $P_{\boldsymbol{\theta}} = \bar{\sigma} \circ f_{\boldsymbol{\theta}}$, so $P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x}) \equiv \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_k$. Thus, the discriminator is represented by a function $P_{\boldsymbol{\theta}} : \mathbb{R}^D \to [0, 1]^K$, which is parametrized by a vector $\boldsymbol{\theta}$.

We will consider a set of popular uncertainty metrics, which we introduced in Sections 2.2.1 and 2.2.2 and restate them here. Firstly, Hendrycks and Gimpel (2017) introduce a simple baseline, which is the highest probability observed for any class, also referred to as confidence:

$$\hat{p} = \max_{k \in [K]} P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x}). \tag{4.1}$$

Ideally, the model's predictive distribution would become more uniform for challenging inputs (e.g. in areas of class overlap) and thus produce a lower confidence score $\hat{p}$, which is why we measure *un*certainty by $1 - \hat{p}$. Another approach lies in measuring the Shannon entropy H of the predictive distribution:

$$\text{H}\big[P_{\boldsymbol{\theta}}(y \mid \mathbf{x})\big] = -\sum_{k=1}^{K} P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x}) \log P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x}). \tag{4.2}$$

The entropy here is minimal when all probability mass is centered on a single class, and maximal when the predictive distribution is uniform. The other uncertainty estimation techniques are based on the idea of Bayesian deep learning, where, the more predictions between different parameter sets disagree, the larger the uncertainty. One straightforward way to measure this disagreement

[46] The following proofs also hold for binary classifiers which are parameterized through a sigmoid function. For the connection between the softmax and sigmoid function, refer to Appendix A.7.

is the average variance of the predicted probability per class, as done in Smith and Gal (2018):

$$\bar{\sigma}^2 = \frac{1}{K}\sum_{K=1}^{K} \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\big[P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})^2\big] - \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\big[P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})\big]^2. \tag{4.3}$$

Maximum softmax and predictive entropy only capture the total uncertainty, and while the class variance aims to quantify model uncertainty, it does so rather heuristically. Thus, we also consider the mutual information between model parameters and a data sample (Depeweg et al., 2018; Smith and Gal, 2018) as a more theoretically-motivated measure of epistemic uncertainty:

$$\underbrace{\mathrm{I}\big[y, \boldsymbol{\theta} \mid \mathbb{D}, \mathbf{x}\big]}_{\text{Model uncertainty}} = \underbrace{\mathrm{H}\Big[\mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\big[P_{\boldsymbol{\theta}}(y \mid \mathbf{x})\big]\Big]}_{\text{Total uncertainty}} - \underbrace{\mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\Big[\mathrm{H}\big[P_{\boldsymbol{\theta}}(y \mid \mathbf{x})\big]\Big]}_{\text{Data uncertainty}}. \tag{4.4}$$

The term itself can be interpreted as the gain in information about the ideal model parameters and correct label upon receiving an input. If we can only gain a little, that implies that parameters are already well-specified and that the epistemic uncertainty is low. Especially when an input is OOD, we therefore expect this metric to display high uncertainty.

4.1.2 Monotonicity & Polytopes

Before developing the main results, we introduce some concepts that will become central to the proofs in the next sections. This includes the definition of unbounded polytopes on which the model behaves linearly, and the monotonicity of multivariate functions, which lets us make statements about the output of the network when scaling its input.

In the univariate case, we call a function strictly increasing on an interval $\mathbb{I} = [a, b]$ with $a < b$ and $a, b \in \mathbb{R}$ if its derivative is strictly positive on the whole interval:

$$\forall x' \in \mathbb{I}: \quad \frac{\partial}{\partial x} f(x)\big|_{x=x'} > 0, \tag{4.5}$$

where $\cdot|_{x=x'}$ refers to evaluating the value of the derivative of f at x'. This definition can also be extended to multivariate functions by requiring strict monotonicity (strictly increasing or decreasing) in all dimensions:

Definition 6 (Monotonicity in Multivariate Functions)**.** We call a multivariate function $f : \mathbb{R}^D \to \mathbb{R}$ strictly monotonic on a subspace $\mathbb{P} \subseteq \mathbb{R}^D$ if it holds that the function is either strictly increasing or decreasing in every direction:

$$\begin{aligned} \forall d \in [D]: \quad & \forall \mathbf{x}' \in \mathbb{P}: \ \big(\nabla_{\mathbf{x}} f(\mathbf{x})\big|_{\mathbf{x}=\mathbf{x}'}\big)_d < 0 \\ \text{or} \quad & \forall \mathbf{x}' \in \mathbb{P}: \ \big(\nabla_{\mathbf{x}} f(\mathbf{x})\big|_{\mathbf{x}=\mathbf{x}'}\big)_d > 0, \end{aligned} \tag{4.6}$$

where $(\cdot)_d$ refers to $\frac{\partial f(x_d)}{\partial x_d}|_{x_d = x'_d}$, i.e. the d-th component of the gradient $\nabla_{\mathbf{x}} f(\mathbf{x})$ evaluated at $\mathbf{x}'$. We call a multivariate function $f : \mathbb{R}^D \to \mathbb{R}^K$ *component-wise strictly monotonic* if the above definition holds for the gradient of every output component $\nabla_{\mathbf{x}} f(\mathbf{x})_k$.

We note here that the softmax function, whose probabilistic output is used for the discussed uncertainty metrics, is an example for a component-wise strictly monotonic function. As later lemmas investigate the behavior of functions in the limit, it is furthermore useful to define regions of the feature space that are unbounded in at least one direction. We call a *partially-unbounded polytope* (henceforth abbreviated by PUP) a convex subspace of $\mathbb{R}^D$ that is unbounded in at least one dimension d, i.e. if the polytope's projection onto d is either left-bounded by $-\infty$ or right-bounded by ∞, or both.

4.1.3 Convergence of Predictions on OOD Data

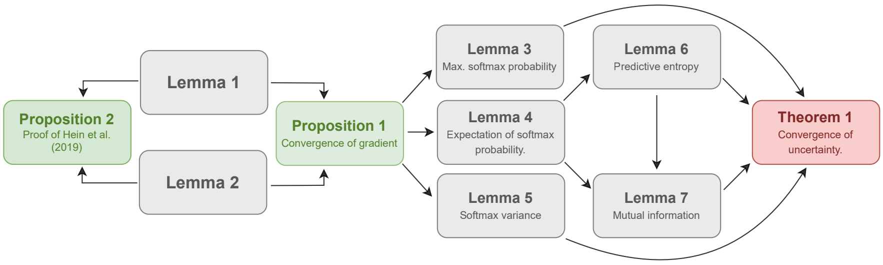


Figure 4.2: Dependencies between theoretical results. Information in parentheses denotes the section in the document.

In this section we will show that, moving the input to the extremes of the feature space, a ReLU classifier will converge to a fixed prediction. To demonstrate this, we must establish how the distance from the training data affects the network's logits. To this end, we utilize a known result stating that neural networks employing piece-wise linear activation functions partition the input space into polytopes (such as in Figure 4.1b; Arora et al., 2018). Given the saturating nature of the softmax, we conclude in Proposition 1 that even for extreme feature values in the limit, the output distribution of the model will not change anymore. In order to help the reader untangle the interdependence of upcoming results, we provide a flow chart in Figure 4.2. We first describe how to re-write a ReLU network—or any other network with piece-wise linear activation functions—as a piece-wise affine transformation, borrowing from Croce and Hein (2018) and Hein et al. (2019). We start with the common form of $f_{\boldsymbol{\theta}}$ as a series of affine transformations, interleaved with ReLU activation functions, which we will denote by ϕ:

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{W}_L\, \phi\big(\,\mathbf{W}_{L-1}\, \phi\big(\ldots\phi\big(\,\mathbf{W}_1\, \mathbf{x} + \mathbf{b}_1\,\big)\ldots\big) + \mathbf{b}_{L-1}\,\big) + \mathbf{b}_L\,. \tag{4.7}$$

In the following, let $f^l_{\boldsymbol{\theta}}(\mathbf{x})$ denote the output of layer l before applying an activation function. We now define a layer-specific diagonal matrix $\Phi_l \in \mathbb{R}^{n_l \times n_l}$ in the following way, where n_l denotes the hidden units in layer l:

$$\Phi_l(\mathbf{x}) = \begin{bmatrix} \mathbb{1}\big(f^l_{\boldsymbol{\theta}}(\mathbf{x})_1 > 0\big) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbb{1}\big(f^l_{\boldsymbol{\theta}}(\mathbf{x})_{n_l} > 0\big) \end{bmatrix}. \tag{4.8}$$

This allows us to rewrite Equation (4.7) by replacing the usage of ϕ with a matrix multiplication using Φ_l:

$$\begin{aligned} f_{\boldsymbol{\theta}}(\mathbf{x}) = {} & \mathbf{W}_L\, \Phi_{L-1}(\mathbf{x})\big(\,\mathbf{W}_{L-1}\, \Phi_{L-2}(\mathbf{x}) \\ & \big(\ldots\Phi_1(\mathbf{x})\big(\,\mathbf{W}_1\, \mathbf{x} + \mathbf{b}_1\,\big)\ldots\big) + \mathbf{b}_{L-1}\,\big) + \mathbf{b}_L\,. \end{aligned} \tag{4.9}$$

We can now distribute the matrix products inside-out, we which demonstrate below using a three-layer network:

$$\begin{aligned} f_{\boldsymbol{\theta}}(\mathbf{x}) &= \mathbf{W}_3\, \Phi_2(\mathbf{x})\big(\,\mathbf{W}_2\, \Phi_1(\mathbf{x})\big(\,\mathbf{W}_1\, \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_3 && (4.10) \\ &= \mathbf{W}_3\, \Phi_2(\mathbf{x})\big(\,\mathbf{W}_2\, \Phi_1(\mathbf{x})\, \mathbf{W}_1\, \mathbf{x} + \mathbf{W}_2\, \Phi_1(\mathbf{x})\, \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_3 && (4.11) \\ &= \underbrace{\mathbf{W}_3\, \Phi_2(\mathbf{x})\, \mathbf{W}_2\, \Phi_1(\mathbf{x})\, \mathbf{W}_1}_{=\ \mathbf{V}(x)}\, \mathbf{x} \\ &\quad + \underbrace{\mathbf{W}_3\, \Phi_2(\mathbf{x})\, \mathbf{W}_2\, \Phi_1(\mathbf{x})\, \mathbf{b}_1 + \mathbf{W}_3\, \Phi_2(\mathbf{x})\, \mathbf{b}_2 + \mathbf{b}_3}_{=\ \mathbf{a}(\mathbf{x})}\,. && (4.12) \end{aligned}$$

This result lets rewrite the network as a single affine transformation $f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{V}(\mathbf{x})\, \mathbf{x} + \mathbf{a}(\mathbf{x})$ with

$$\mathbf{V}(\mathbf{x}) = \mathbf{W}_L \left(\prod_{l=1}^{L-1} \Phi_l(\mathbf{x})\, \mathbf{W}_{L-l} \right) \tag{4.13}$$

$$\mathbf{a}(\mathbf{x}) = \mathbf{b}_L + \sum_{l=1}^{L-1} \left(\prod_{l'=1}^{L-l} \mathbf{W}_{L+1-l'}\, \Phi_{L-l'}(\mathbf{x}) \right) \mathbf{b}_l \,. \tag{4.14}$$

Note that the definition of $\mathbf{V}(\mathbf{x})$ corresponds to the Jacobian of $f_{\boldsymbol{\theta}}(\mathbf{x})$, meaning that $v_{kd} = \frac{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_k}{\partial x_d}$. This is very useful, as it allows us to quickly check whether a network $f_{\boldsymbol{\theta}}$ is component-wise strictly monotonic by checking $\mathbf{V}(\mathbf{x})$ for entries containing zeros. As Hein et al. (2019) show, this formulation can also be used to characterize a set of polytopes $\mathcal{Q} = \{Q_1, \ldots, Q_M\}$ induced by $f_{\boldsymbol{\theta}}$ and that within each polytope, the function has a unique representation as an affine transformation. For this reason, we drop the dependence of $\mathbf{V}$ and $\mathbf{a}$ on $\mathbf{x}$ when we refer to a specific polytope. Such polytopes are constructed by first retrieving the half-spaces induced by each of the network's neurons and then intersecting all said half-spaces to generate convex regions or polytopes.[47] We are especially interested in polytopes that are unbounded in at least one direction. The results of Croce and Hein (2018) and Hein et al. (2019) show that there is a finite number of polytopes corresponding to the given network, and their Lemma 3.1 proves the existence of at least one unbounded polytope. Furthermore, under a mild condition on $\mathbf{V}$, we can ascertain that $f_{\boldsymbol{\theta}}$ will be component-wise strictly monotonic on any polytope.

Lemma 1. *Suppose $f_{\boldsymbol{\theta}}$ is a ReLU network according to Definition 5. Then $f_{\boldsymbol{\theta}}$ is a component-wise strictly monotonic function on every of its polytopes $Q \in \mathcal{Q}$, as long as its corresponding matrix $\mathbf{V}$ has no zero entries.*

Proof. Let Q be one such polytope. As discussed, when restricted to Q, the network corresponds to an affine transformation $f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{V}\,\mathbf{x} + \mathbf{a}$ with $\mathbf{V} \in \mathbb{R}^{K \times D}$ and $\mathbf{a} \in \mathbb{R}^K$. $f_{\boldsymbol{\theta}}(\mathbf{x})_k$ thus corresponds to the dot product of the k-th row of $\mathbf{V}$ and $\mathbf{x}$ plus the k-th element of $\mathbf{a}$. It follows that the partial derivative of $f_{\boldsymbol{\theta}}(\mathbf{x})_k$ with respect to a dimension d equals the element v_{kd} in $\mathbf{V}$. This entails that, if $v_{kd} \neq 0$, at any point $\mathbf{x} \in Q$ the gradient will be always positive or always negative. □

[47] We refer the reader to Appendix A.8 or Hein et al. (2019) for details on the construction, since it is not central to our reasoning.

We note here that the component-wise strict monotonicity of $f_{\boldsymbol{\theta}}$ and softmax do not entail the same property for $P_{\boldsymbol{\theta}}$.[48] Nonetheless, the monotonic behavior of $f_{\boldsymbol{\theta}}$ is sufficient to drive the logits to plus or minus infinity in the limit, a phenomenon that constrains the output of $p_{\boldsymbol{\theta}}$ as we scale a data sample away from training data. We begin our investigation of behavior in the limit by showing that if we scale a vector only in a single dimension, we eventually always remain within a unique PUP.

Lemma 2. *Let $\mathbf{x}' \in \mathbb{R}^D$ and $\mathcal{Q} = \{Q_1, \ldots, Q_M\}$ be the finite set of polytopes generated by a network $f_{\boldsymbol{\theta}}$. Let $\boldsymbol{\alpha} \in \mathbb{R}^D$ be a vector s.t. $\forall d' \neq d, \alpha_{d'} = 1$. There exist a value $\beta > 0$ and $m \in 1, \ldots, M$ such that for all $\alpha_d > \beta$, the product $\mathbf{x}' \circ \boldsymbol{\alpha}$ lies within Q_m.*

Proof. The proof mirrors the proof of Lemma 3.1 in Hein et al. (2019), so we only provide the intuition. By contradiction, suppose that there is no unique polytope and thus the point $\mathbf{x}' \circ \boldsymbol{\alpha}$ must traverse different polytopes as we scale up α_d. Since there are finitely many polytopes, eventually the same polytope Q_m will have to be traversed twice. Since the polytopes are convex, all the points on the line connecting the locations of where the boundary of Q_m was crossed the first and second time must lie within Q_m, but this contradicts the fact that the scaled point traverses different polytopes. □

From here onward, we adapt the following shorthand to simplify notation: Given a scaling vector $\boldsymbol{\alpha} \in \mathbb{R}^D$ s.t. $\forall d' \neq d, \alpha_{d'} = 1$, we use $\mathbb{P}(\mathbf{x}', d)$ to denote the PUP that $\mathbf{x}'$ lands in when scaling it with α_d in the limit. This definition implies that we can only scale parallel to the basis vectors and not arbitrary directions (for a discussion on how restrictive this is, see Section 4.1.5). Finally, in the next lemma we establish that the output distribution converges to a fixed point using the l_2-norm of the gradient $\nabla_{\mathbf{x}} P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})$. Generally, in regions of the feature space where the classifier predicts the same probability distribution over classes, small perturbations in the input $\mathbf{x}$ will not change the prediction. Therefore, the gradient in these regions w.r.t. the input will be small and potentially even correspond to the zero vector, with a norm of (or close to) zero.

Proposition 1 (Convergence of predictions in the limit)**.** *Suppose that $f_{\boldsymbol{\theta}}$ is a ReLU-network. Let $\mathbf{x}' \in \mathbb{R}^D$, suppose $\boldsymbol{\alpha}$ is a scaling*

[48] To see a counterexample, the reader can check that even assuming component-wise strict monotonicity for $f_{\boldsymbol{\theta}}$, if the matrix $\mathbf{V}$ associated to $f_{\boldsymbol{\theta}}$ on a specific polytope has a column d filled with the same value a, then the resulting $p_{\boldsymbol{\theta}}$ will have a gradient of 0 at dimension d, regardless of what class we are considering. This is because the partial derivatives of the softmax, when all multiplied by the same constant a, add up to zero.

vector and that the associated PUP $\mathbb{P}(\mathbf{x}', d)$ has a corresponding matrix $\mathbf{V}$ with no zero entries. Then it holds that

$$\forall k \in [K] : \quad \lim_{\alpha_d \to \infty} \Big|\Big|\nabla_{\mathbf{x}} P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})\big|_{\mathbf{x}=\boldsymbol{\alpha} \circ \mathbf{x}'}\Big|\Big|_2 = 0. \qquad (4.15)$$

The whole proof can be found in Appendix A.9, so we present the main intuitions here. Because of Lemma 2, we know the scaled point $\boldsymbol{\alpha} \circ \mathbf{x}'$ will end up in a unique PUP. The assumption on $f_{\boldsymbol{\theta}}$ then triggers Lemma 1, from which we can infer that scaling the input in a single dimension leads all logits to $\pm\infty$. Because of the saturating property of the softmax, this will in turn provoke the output of $p_{\boldsymbol{\theta}}$ to converge to a fixed point. As an aside, we recast Theorem 3.1 by Hein et al. (2019) in our framework, showing that the model becomes increasingly certain in a single class, placing all its probability mass on it in the limit. The proof of this additional proposition is in Appendix A.10.

Proposition 2. *Let $f_{\boldsymbol{\theta}}$ be ReLU network. Let $\mathbf{x}' \in \mathbb{R}^D$, suppose $\boldsymbol{\alpha}$ is a scaling vector and that the associated PUP $\mathbb{P}(\mathbf{x}', d)$ has a corresponding matrix $\mathbf{V}$ with no zero entries. Assume the d-th column of $\mathbf{V}$ has no duplicate entries. Then there exists a class k such that*

$$\lim_{\alpha_d \to \infty} \bar{\sigma}(f_{\boldsymbol{\theta}}(\boldsymbol{\alpha} \circ \mathbf{x}'))_k = 1.$$

In conclusion, we have shown in this section that the output probabilities of ReLU networks are less and less sensitive to small perturbations of the input in the limit and, under the assumptions of Proposition 2, will converge to favor a single class with very high confidence. In the next section we prove that all other uncertainty metrics also converge to fixed values in the limit.

4.1.4 Convergence of Uncertainty Metrics on OOD Data

In Proposition 1, we have established how the prediction of a model converges to a fixed point when feature values become extreme. We now show a similar property about the uncertainty estimation techniques introduced in Section 4.1.1. The fact that this the same pathologies appear for more complex metrics is not immediately obvious, and one might assume that we can curb the deficiency of the simple confidence score by using more sophisticated metrics and Bayesian deep learning techniques. To this end, we have to establish how the predictions coming from multiple model instances interact, a point we analyze in Lemma 4. Then, we demonstrate how the uncertainty metrics also converge to a fixed value in the limit by proving the case for each of them in turn, before bundling our results in Theorem 1. We start with the easiest metric, which also applies to a single ReLU network.

Lemma 3 (Maximum softmax probability)**.** *Suppose that $f_{\boldsymbol{\theta}}$ is a ReLU network. Let $\mathbf{x}' \in \mathbb{R}^D$, suppose $\boldsymbol{\alpha}$ is a scaling vector and that the associated PUP $\mathbb{P}(\mathbf{x}', d)$ has a corresponding matrix $\mathbf{V}$ with no zero entries. Then*

$$\lim_{\alpha_d \to \infty} \big|\big|\nabla_{\mathbf{x}} \max_{k \in [K]} P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})\big|_{\mathbf{x}=\boldsymbol{\alpha} \circ \mathbf{x}'}\big|\big|_2 = 0.$$

Proof. The gradient of the max function will be a specific $\nabla_{\mathbf{x}} P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})$, which reduces this to the case already proven in Proposition 1. □

Note that for this metric, the combination with Proposition 2 shows that the model is fully confident in a single class in the limit. For our following lemmas, we consider uncertainty scores that are based on multiple instances, e.g. different ensemble members or forward passes using re-sampled dropout masks. What all of these approaches have in common is that for every b in $1, \ldots, B$, the network parameters $\boldsymbol{\theta}^{(b)}$ will differ, and thus also the polytopal tesselation of the feature space. Hence, we have to adjust our assumptions accordingly. For every instance b, let us denote the affine function on a polytope $Q^{(b)}$ as $f_{\boldsymbol{\theta}}^{(b)}(\mathbf{x}) = \mathbf{V}^{(b)}\mathbf{x} + \mathbf{a}^{(b)}$. In order for our previous strategy to hold, we now assume for all $b \in [B]$ that $\mathbb{P}^{(b)}(\mathbf{x}', d)$ has a matrix $\mathbf{V}^{(b)}$ which does not have any zero entries. Note that even though this assumption has to hold for all b, this does not mean that the matrices have to be identical.

Lemma 4 (Convergence of aggregated predictions in the limit)**.** *Suppose that $f_{\boldsymbol{\theta}}^{(1)}, \ldots, f_{\boldsymbol{\theta}}^{(K)}$ are ReLU networks. Let $\mathbf{x}' \in \mathbb{R}^D$, suppose $\boldsymbol{\alpha}$ is a scaling vector and that for all k, the associated PUP $\mathbb{P}^{(k)}(\mathbf{x}', d)$ has a corresponding matrix $\mathbf{V}^{(k)}$ with no zero entries. Then*

$$\lim_{\alpha_d \to \infty} \big|\big|\nabla_{\mathbf{x}}\ \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\big[P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})\big]\big|_{\mathbf{x}=\boldsymbol{\alpha} \circ \mathbf{x}'}\big|\big|_2 = 0.$$

The full proof of this lemma can be found in Appendix A.11. The analogous lemmas for the remaining uncertainty metrics—predictive entropy, class variance and mutual information—are stated and proved in Appendices A.12 to A.14. The proof strategy for all further metrics is to simplify and reduce the uncertainty metrics such that Lemma 4 or Proposition 1 can be applied. All of these results combined now pave the way for our central theorem.

Theorem 1 (Convergence of uncertainty level in the limit)**.** *Suppose that $f_{\boldsymbol{\theta}}^{(1)}, \ldots, f_{\boldsymbol{\theta}}^{(B)}$ are ReLU networks. Let $\mathbf{x}' \in \mathbb{R}^D$, suppose $\boldsymbol{\alpha}$ is a scaling vector and that for all b, the associated PUP $\mathbb{P}^{(b)}(\mathbf{x}', d)$ has a corresponding matrix $\mathbf{V}^{(b)}$ with no zero entries. Then, whenever uncertainty is measured via either of the following metrics*

1. *Maximum softmax probability in Equation (4.1);*
2. *Predictive entropy in Equation (4.2);*
3. *Class variance in Equation (4.3);*
4. *Approximate mutual information in Equation (4.4);*

the network(s) will converge to fixed uncertainty scores for $\mathbf{x}' \circ \boldsymbol{\alpha}$ in the limit of $\alpha_d \to \infty$.

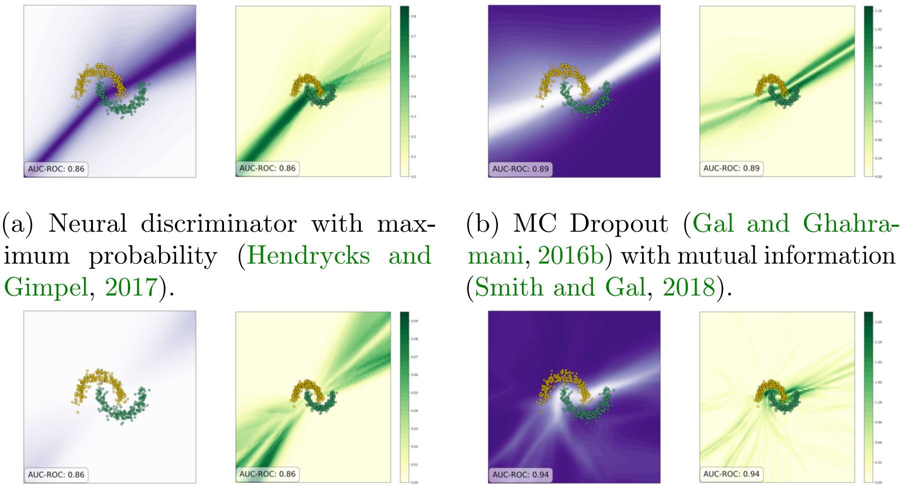
Proof. The four parts of the theorem are proven separately by Lemmas 3 and 7 to 9 and Appendix A.12 in Appendix A. □

What follows from this result is that methods based on multiple instances of ReLU classifiers will suffer from the aforementioned problem as long as uncertainty is estimated with one of the techniques listed above. Next we demonstrate how these assumptions and results apply on synthetic data.

4.1.5 Synthetic Data Experiments

To illustrate our findings, we plot the uncertainty surfaces and the gradient magnitudes of different models and uncertainty metric pairings on the half moons dataset, which we generate using the corresponding function in the `scikit-learn` package (Pedregosa et al., 2011). Detailed information about the procedure can be found in Appendix C.4.2 along with additional plots.

For a single network, we can observe in Figure 4.3a that there exist vast open-ended regions of stable confidence, confirming the findings of Theorem 1. However, in the right part of Figure 4.3a we can observe green regions with high gradient magnitude which do not seem to comply with our findings. In this case, we can see that these regions follow the decision boundaries. Due to the exponential function in the softmax, it is intuitive that small perturbation in these areas would have a large impact on the uncertainty score, resulting in a high gradient magnitude. But why does the magnitude not decrease in the limit as predicted by Theorem 1? We formulated our scaling vector $\boldsymbol{\alpha}$ in way that only allows scaling along one of the coordinate axes. Therefore, if



(a) Neural discriminator with maximum probability (Hendrycks and Gimpel, 2017).

(b) MC Dropout (Gal and Ghahramani, 2016b) with mutual information (Smith and Gal, 2018).

(c) Neural ensemble (Lakshminarayanan et al., 2017) with class variance.

(d) Anchored ensemble (Pearce et al., 2020) with mutual information (Smith and Gal, 2018).

Figure 4.3: Uncertainty on the half-moon dataset, including the binary classification AUROC. (Left plots) The uncertainty surface is represented with increasingly darker shades of purple, with white being the lowest uncertainty. Open-ended regions of static certainty appear across different models and metrics, and are extrapolated to unseen data (see Figures 4.3a to 4.3d); this phenomenon is less apparent in some instances (Figure 4.3d). (Right plots) Increasing shades of green indicate the magnitude of the gradient of the uncertainty score w.r.t. the input. All metrics show open ended regions where the magnitude approaches zero.

the decision boundaries are not parallel to the axes, by scaling we eventually escape the green areas and arrive at an area with gradient of magnitude zero. If the green regions were parallel to the axes then this would result in a violation of our main assumption. Traversing the input space parallel to a decision boundary in direction d will not influence the prediction within the polytope, meaning that there will be entries $v_{cd} = 0$.[49]

Turning to predictions aggregated from multiple network instances in Figures 4.3b to 4.3d, we again observe large regions of constant uncertainty. The high-confidence region in the plots using mutual information (Figure 4.3d) displays a different behavior from the others. As this metric aims to isolate epistemic uncertainty, it makes sense that uncertainty would be lowest around the training

[49] A decision boundary in a polytope is not the only way in which this assumption can be broken, but it still appears to hold reasonably often. For instance, just around 6.3% of plotted points in Figure 4.1 possess a matrix $\mathbf{V}$ with at least one zero entry—all located in the PUP in the top right corner.

data, i.e. where the model is best specified. The character of the green regions in the bottom part of Figure 4.3c and Figure 4.3d can again be explained by decision boundaries: In these cases, we have multiple instances with parameters $\boldsymbol{\theta}^{(k)}$, all with their own polytopal structure. When they overlap, the regions of the feature space where the assumption of our theorem is violated can either grow (Figure 4.3c) or shrink (Figure 4.3d), depending on the diversity among instances. The fact that the anchored ensemble in Figure 4.3d does not exhibit such uniform regions of uncertainty like the vanilla ensemble could be explained by the fact that its training procedure encourages diversification between members. The difference between MC Dropout and ensemble models can be elucidated using recent insights that variational methods tend to only explore a single mode of the weight posterior $p(\boldsymbol{\theta} \mid \mathbb{D})$, while ensemble members often spread across multiple modes (Wilson and Izmailov, 2020).

Overall, we have seen that our theorem can explain why an overgeneralization of uncertainty scores beyond the training data results in failure in OOD detection. We also explored the cases in which our assumptions are violated, i.e. by multiple, diverse model instances. In such scenarios, identification of OOD samples could in theory succeed, but often fails to do so reliably, see e.g. Snoek et al. (2019); Ulmer et al. (2020). These insights can also help explain many other empirical findings in this regard on a variety of real-world datasets, e.g. Smith and Gal (2018); **?**.

4.2 Uncertainty & Calibration in Low-Resource NLP

The following work is based on Ulmer et al. (2022b).

The previous section looked at a somewhat simplified setting using ReLU networks. In practice, most contemporary NLP architectures are based on much more complex architectures like the transformer (Vaswani et al., 2017). Theoretical arguments like Theorem 1 then become harder, since making monotonicity arguments with model components such as multi-head attention is not trivial. Additionally, the proof strategy of scaling a single feature value into the limit is not applicable, because the input changes from single feature vectors to a series of subword token embeddings. For this reason, we turn to an empirical approach instead.

While there exist many works on images (Lakshminarayanan et al., 2017; Snoek et al., 2019) and tabular data (Ruhe et al., 2019; Ulmer et al., 2020; Malinin et al., 2021), the quality of uncertainty estimates provided by neural networks remains underexplored in NLP. In addition, as model underspecification due to insufficient data presents a risk (D'Amour et al., 2022), the increasing interest in less-researched languages with limited resources raises the question of how reliably uncertain predictions can be identified. This motivates the following research questions:

1. What are the best approaches in terms of uncertainty quality and calibration?
2. How are models impacted by the amount of available training data?
3. What are differences in how the different approaches estimate uncertainty?

Contributions. We address these questions by conducting a comprehensive empirical study of eight different models for uncertainty estimation for classification and evaluate their effectiveness on three languages spanning distinct NLP tasks, involving sequence labeling and classification. We show that while approaches based on pre-trained models and ensembles achieve the best results overall, the quality of uncertainty estimates on OOD data can become worse using *more* data. In a qualitative analysis, we also discover that a model's total uncertainty seems to mostly consist of its data uncertainty.

4.2.1 Methodology

Models. We choose a variety of models that cover a range of different approaches based on the two most prominently used architectures in NLP: Long-short term memory networks (LSTMs; Hochreiter and Schmidhuber, 1997) and transformers (Vaswani et al., 2017). Inside the first family, we use the variational LSTM (Gal and Ghahramani, 2016a) based on MC dropout (Gal and Ghahramani, 2016b), the Bayesian LSTM (Fortunato et al., 2017) implementing Bayes-by-backprop (Blundell et al., 2015) and the ST-τ LSTM (Wang et al., 2021a), modeling transitions in a finite-state automaton, as well as an ensemble (Lakshminarayanan et al., 2017). In the second family, we count the variational transformer (Xiao et al., 2020), also using MC dropout, the SNGP transformer (Liu et al., 2023), using a Gaussian Process output layer, and the deep deterministic uncertainty transformer (DDU; Mukhoti et al., 2021), fitting a Gaussian mixture model on extracted features. We elaborate on implementation details in Appendix C.6.

Uncertainty Metrics. We test the same metrics as introduced in Section 4.1.1, but add a few additional ones. One of them is the softmax gap (Tagasovska and Lopez-Paz, 2019), i.e. the difference between the two largest probabilities of the classifier's output distribution. As another metric, we consider the Dempster-Shafer metric (Sensoy et al., 2018), defined as $K/(K + \sum_{k=1}^{K} \exp(z_k))$, where z_k denotes the logit corresponding to class k. Since this metric considers logits, it might be able to avoid the saturation on OOD shown by Hein et al. (2019) or in Section 4.1.4. While all metrics so far can be mixed and matched with all the tested models, there are also a few model-specific metrics. For instance, the DDU transformer by Mukhoti et al. (2021) uses the log-probability of the last layer network activation under a Gaussian mixture model fitted on the training set as an additional metric. Since all others models are trained or fine-tuned as classifiers, they cannot assign log-probabilities to sequences. Lastly, since some tasks require predictions for every time step of a sequence, we determine the uncertainty of a whole sequence in these cases by taking the mean over all step-wise uncertainties.[50] A more principled approach for sequences is for instance provided by Malinin and Gales (2021) in the context of NLG, and we leave the extension and exploration of such methods for different uncertainty metrics, models and tasks to future work.

4.2.2 Dataset Selection & Creation

Lang.	Task	Dataset	OOD Test Set	# ID / OOD	Training Sizes
EN	Intent Classification	Clinc Plus (Larson et al., 2019)	Out-of-scope voice commands	15k/1k	15k/12.5k/10k
DA	Named Entity Recognition	Dan+ News (Plank et al., 2020)	Tweets	4382/109	4k/2k/1k
FI	PoS Tagging	Finnish UD Treebank (Haverinen et al., 2014; Pyysalo et al., 2015; Kanerva and Ginter, 2022)	Hospital records, online forums, tweets, poetry	12217/2122	10k/7.5k/5k

Table 4.1: Datasets used for our experiments. The original and subsampled number of sequences for experiments are given on the right.

[50] We also just considered the *maximum* uncertainty over a sequence, with similar results.

In-Distribution Training Sets. In our experiments, we test three different languages combined with one NLP task, each. For the languages, we choose English (Clinc Plus; Larson et al., 2019), Danish in the form of the Dan+ dataset (Plank et al., 2020) based on news texts from PAROLE-DK (Bilgram and Keson, 1998), Finnish (UD Treebank; Haverinen et al., 2014; Pyysalo et al., 2015; Kanerva and Ginter, 2022). These datasets correspond to the NLP tasks of sequence classification, named entity recognition and part-of-speech tagging, respectively. An overview over the datasets is given in Table 4.1, with the preprocessing detailed in Appendix C.5. We use low-resource languages in the case of Finnish and Danish, and simulate a low-resource setting using English data.[51] Starting with a sufficiently-sized training set and then sub-sampling allows us to create training sets of arbitrary sizes. We employ a specific sampling scheme that tries to maintain the sequence length and class distribution of the original corpus, which we explain and verify in Appendix B.3.

Out-Of-Distribution Test Sets. We create OOD test sets from data sources that are qualitatively different from the in-distribution training data: Out-of-scope voice commands by users in Larson et al. (2019),[52] the Twitter split of the Dan+ dataset (Plank et al., 2020), and the Finnish OOD treebank (Kanerva and Ginter, 2022). In similar works for the image domain, OOD test sets are often chosen to be convincingly different from the training distribution, for instance MNIST versus Fashion-MNIST (Nalisnick et al., 2019b; van Amersfoort et al., 2021). While there exist a variety of taxonomies for distributional shifts(Moreno-Torres et al., 2012; Wald et al., 2021; Arora et al., 2021; Federici et al., 2021; Hupkes et al., 2023), it is often hard to determine if and what kind of shift is taking place. Winkens et al. (2020) define *near OOD* as a scenario in which the training and outlier distribution are meaningfully related, and *far OOD* as a case in which they are unrelated. Unfortunately, this distinction is somewhat arbitrary and hard to apply in a language context, where OOD *could* bde

[51] The definition of low-resource actually differs greatly between works. One definition by Bird (2022) advocates the usage for (would-be) standardized languages with a large amount of speakers and a written tradition, but a lack of resources for language technologies. Another way is a task-dependent definition: For dependency parsing, Müller-Eberstein et al. (2021) define low-resource as providing less than 5000 annotated sentences in the Universal Dependencies Treebank. Hedderich et al. (2021); Lignos et al. (2022) lay out a task-dependent spectrum, from a several hundred to thousands of instances.

[52] Since all instances in this test set correspond to out-of-scope inputs and not to classes the model was trained on, we cannot evaluate certain metrics in Table 4.2.

defined as anything ranging from a different language or dialect to a different demographic of an author or speaker or a new genre. Therefore, we use a similar methodology to the validation of the sub-sampled training sets to make an argument that the selected OOD splits are sufficiently different in nature from the training splits. The exact procedure along some more detailed results is described in Appendix B.4. Mainly, we examine the distribution of sequence lengths and labels, and score the OOD test set using the perplexity of a language model training on the training split.

4.2.3 Model Training

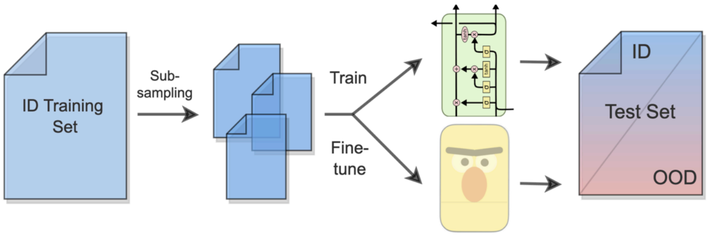


Figure 4.4: Schematic of our text classification experiments. Training sets are sub-sampled and used to train LSTM-based models and fine-tune transformer-based ones, which are evaluated on in- and out-of-distribution test data.

Unfortunately, our datasets do not contain enough data to train transformer-based models from scratch. Therefore, we only fully train LSTM-based models, and use pre-trained transformers, namely Bert (English; Devlin et al., 2019), Danish Bert (Hvingelby et al., 2020), and FinBert (Finnish; Virtanen et al., 2019), for the other approaches. The whole procedure is depicted in Figure 4.4. The way we optimize models is provided in **??**. We list training hardware, hyperparameter information in Appendix C.4.3, with the environmental impact described in Appendix C.2.

4.2.4 Evaluation

In addition to evaluating models on the task performance, we also evaluate the following calibration and uncertainty, painting a multi-faceted picture of the reliability of models. In all cases, we use the almost stochastic order test (ASO; del Barrio et al., 2018a; Dror et al., 2019) as described in Section 3.2.1 for significance testing.

Evaluation of Calibration. First, we measure the calibration of models using the expected calibration error (ECE; Naeini et al., 2015; Guo et al., 2017), which we already discussed in Section 2.2.1. In the same chapter, we introduced the frequentist measure of coverage (Larry, 2004; Kompa et al., 2021). Coverage here based on the (non-conformalized) prediction set of a classifier given an input, which includes the most likely classes adding up to or surpassing $1 - \alpha$ probability mass. A well-tuned classifier should contain the correct class in this prediction set, while minimizing its width. The extent to which this property holds can be determined by the *coverage percentage*, i.e. the number of times the correct class in indeed contained in the prediction set, and its cardinality, denoted simply as *width*.

Evaluation of Uncertainty. We compare uncertainty scores on the ID and OOD test set and measure the area under the receiver-operator curve (AUROC; evaluating the trade-off between sensitivity and specificity) and under the precision-recall curve (AUPR), assuming that uncertainty will generally be higher on samples from the OOD test set.[53] An ideal model should create very different distributions of confidence scores on ID and OOD data, thus maximizing AUROC and AUPR (as opposed to the saturating confidence scores that we observed in Section 4.1.5). However, we also want to find out to what extent uncertainty can give an indication of the correctness of the model, which is why we propose a new way to evaluate the *discrimination* property proposed by Alaa and van der Schaar (2020) based on Leonard et al. (1992): A good model should be less certain for inputs that incur a higher loss. To measure this both on a token and sequence level, we utilize Kendall's τ (Kendall, 1938), which, given two lists of measurements, determines the degree to which they are *concordant*—that is, to what extent the rankings of elements according to their measured values agree. This is expressed by a value between -1 and 1, with the latter expressing complete concordance. In our case, these measurements correspond to the uncertainty estimate and the actual model loss, either for tokens (Token τ) or sequences (Sequence τ).

4.2.5 Experiments

We present the results from our experiments using the largest training set sizes per dataset in Table 4.2.[54]

Task Performance. Across datasets and models, we can identify several trends: some of the Bert-based models unsurprisingly perform better than LSTM-based models, which can be explained by the fact that they are pretrained on large datasets. We observe worse performance for some LSTM and Bert-variants, in particular the variational, Bayesian and ST-τ LSTM, as well the SNGP Bert. In accordance with the ML literature (see e.g. Lakshminarayanan et al., 2017; Snoek et al., 2019) and the discussions in Section 2.2.2, LSTM ensembles actually perform very strongly and on par or sometimes better than fine-tuned Berts.

Calibration. We also see that Bert models generally achieve lower calibration errors across all metrics measured, which is in line with previous works (Desai and Durrett, 2020; Dan and Roth, 2021). It is interesting that the correct prediction is almost always contained in the 0.95 confidence set across all models, however these number have to be interpreted in the context of the set's width: It becomes apparent that for instance LSTMs achieve this coverage by spreading probability mass over many classes, while only Bert-based models, LSTM ensembles as well as the Bayesian LSTM (on Danish) and the Variational LSTM (on Finnish) are *confidently* correct.

Uncertainty Quality. LSTM-based model seem to struggle to distinguish in- from out-of-distribution data based on predictive uncertainty. For Danish, only Berts perform visibly above chance-level. For Finnish, the AUPR results suggest that although some OOD instances are quickly identified as uncertain, many other OOD inputs remain undetected among in-distribution samples. For English, OOD samples are detected more effectively, which can be explained by them consisting of unknown voice commands, representing a potential instance of *semantic* shift, which has been shown to be easier to detect by classifiers (Arora et al., 2021). Furthermore, it is striking that uncertainty and loss on a token-

[53] We thus formulate a pseudo-binary classification task as common in the literature, using the model's uncertainty score to try to distinguish the two test sets. Note that we do not advocate for actually using uncertainty for OOD detection, but only use it for evaluation purposes, since uncertainty on OOD examples should be high due to model uncertainty.

[54] For English, some models were omitted due to convergence issues, which are discussed in Appendix C.7.

level (Token τ) is only positive correlated for some models, using metrics such as the maximum probability score, softmax gap or the Dempster-Shafer metric, which are all entirely based on the categorical output distributions. On a sequence-level (Sequence τ), the correlation is often *negative*, meaning that higher uncertainty goes hand in hand with a *higher* loss. This is the antithesis of the desired outcome and the opposite of the trend on the token-level, and suggests that few tokens-level scores distort the sequence-level aggregation of uncertainties. Lastly, it should be noted that different uncertainty metrics yield diverse outcomes: There does not seem to be one superior metric across all experimental settings, as seen by the variety of markers shown in Table 4.2, which signify the best-performing uncertainty metrics per model and result.

4.2.6 Dependence on Training Data

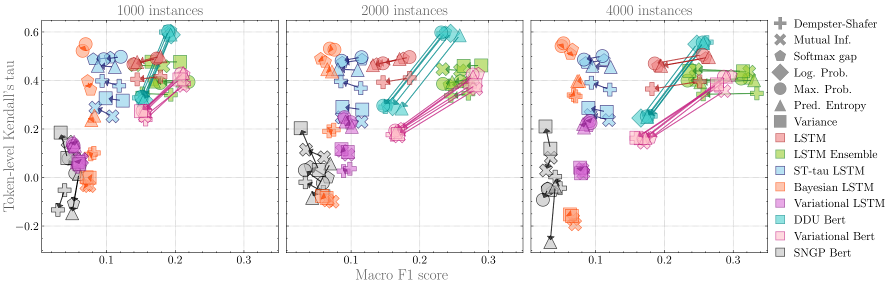
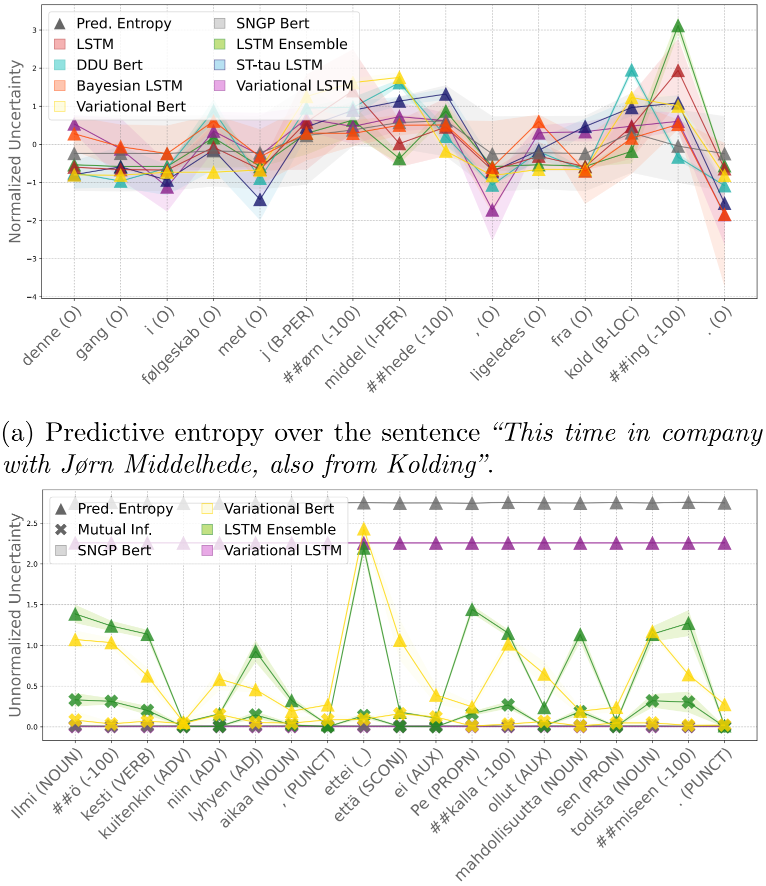


Figure 4.5: Scatter plot showing the difference between model performance (measured by macro F_1 and the quality of uncertainty estimates on a token-level (measured by Kendall's τ). Shown are different models and uncertainty metrics and several training set sizes of the Dan+ dataset. Arrows indicate changes between the in-distribution and out-of-distribution test set. Best viewed electronically and in color.

After presenting the best results for the biggest training set sizes in Table 4.2, we now continue to analyze the difference between models and metrics in a more fine-grained way. In Figure 4.5, we show differences for the token-level correlation between a model's loss and its uncertainty measured by Kendall's τ, with arrows indicating the shift from measurements on the in- to the out-of-distribution test set. Here, we see the same trend of more training data having a larger influence on Bert models. Peculiarly, we also observe that the uncertainty of pre-trained models correlates less with their losses on the OOD data, while this property stays relative constant for LSTMs. We can recognize this trend also for the other datasets in Figure 4.5 and to a lesser degree on a sequence level Figure B.14a in Appendix B.5, albeit with a *negative* correlation in general in the latter case. In Figures B.11 and B.12 in Appendix B.5, we show the AUROC and AUPR of different model-uncertainty metric combinations for all datasets and training set sizes. In both cases, we can notice that pre-trained models profit more from an increase in available training data than LSTM-based models that are trained from scratch. This improvement is observed both in task performance, as well as in the model's ability to discern ID from OOD data using its uncertainty, but more so for the Danish than English or Finnish. Like in the previous section, we often see that uncertainty metrics of the same model perform quite similarly. These results outline a seeming paradox: Pre-trained and then fine-tuned models (often) perform better on the task at hand, and provide better uncertainty estimates, but only on in-distribution data. Models trained from scratch that have seen less data overall, however provide more reliable uncertainty estimates on OOD data, but are also worse calibrated (Section 4.2.5), with the exception of ensembles. This effect appears to largest on Danish, containing the least data.

4.2.7 Qualitative Analysis

We investigate the development of uncertainty estimates over the course of a single sequence for different datasets, models, and uncertainty metrics. Two examples are showcased in Figure 4.6, with more examples in Appendix B.6. By looking at the predictive entropy of models in Section 4.2.7, we can observe multiple things: First of all, we can observe some degree of agreement between models and their uncertainty: Uncertainty is higher for subword tokens, and the total uncertainty always appears to reduce considerably on punctuation. Interestingly, the highest uncertainty seems to be produced by the DDU and variational Bert models as well as the ensembles. In Figure 4.6b, we compare the estimates for predictive entropy and mutual information, the latter of which is supposed to only express model uncertainty. Here, uncertainty is generally low, indicating a large part of the total uncertainty might actually be of an aleatoric nature (which is the gap between triangle and cross markers of the same color, due to Equation (4.4)). These insights indicate that while aleatoric uncertainty might be a constant factor for all models, epistemic uncertainty expectedly differs noticeably between them. We use all of these insights to discuss the choice of model next.



(a) Predictive entropy over the sentence *"This time in company with Jørn Middelhede, also from Kolding"*.

(b) Predictive entropy and mutual information over the sentence *"However, the phenomenon lasted for such a short time that Pekka did not have a chance to prove it"*.

Figure 4.6: Uncertainty estimates on single sequences, for (a) predictive entropy of different models on Danish and (b) predictive entropy and mutual information for multi-prediction models on Finnish (Figure 4.6b).

4.2.8 Discussion

Our experiments in the previous sections have uncovered interesting nuances about uncertainty quantification in text classification. With respect to the first research question, we observed that fine-tuning Berts and training LSTM ensembles on different languages produces high task scores with low calibration errors and high-quality uncertainty estimates, but only on in-distribution data. On OOD data, uncertainty estimates from fine-tuned models actually become less indicative of potential model loss compared to LSTM-based models. We also find that among the variety of uncertainty metrics proposed, there does not appear to be a superior metric, i.e. most able to hint at mispredictions and OOD data. Differences in Kendall's τ on a token and sequence level suggest that loss and uncertainties fluctuate over the course of sequence.

Answering the second research question, more training data paradoxically decreases the quality of uncertainty estimates on OOD data for pre-trained models. We speculate that fine-tuning models increasingly lets them forget relevant features that would produce higher uncertainty. This might explain why for this effect is smaller for LSTM-type models, which are trained from scratch.

Lastly, we conclude about the third research question that all the total uncertainty of models behaves somewhat similarly, potentially due to the strong influence of aleatoric uncertainty. From these insights, we summarize that the approaches using pre-trained models overall give the best trade-off between task performance, uncertainty quality and calibrations, however their failure on OOD samples opens up further directions of research. Ensembles can provide an alternative here in data-scarce settings, when the task is sufficiently learnable without the need for pre-training.

Limitations. Even though the experiments test a large array of models and metrics, the collection here shown is by no means exhaustive, and only a selection of popular models or approaches from very different families were considered. Another glaring shortcoming is the focus on only three European languages: By comparing members of the Uralic, North Germanic and West Germanic families, we only scratch the surface when it comes to the morphological diversity of human language, as for instance illustrated in Figure 1.1c. Further, we only focused on languages with a Latin writing systems, as well as specific text domains and tasks. This is due to resource constraints and the availability of suitable OOD test sets. We hope that follow-up works will refine our insights on a more representative sample of natural languages.

4.3 Summary

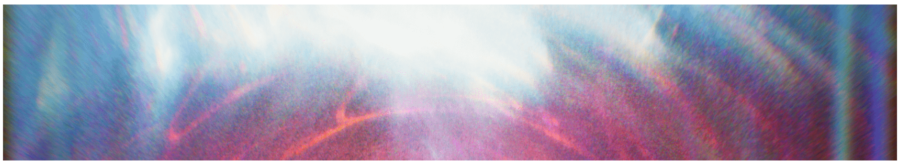
This chapter explored some perspectives on uncertainty quantification in classification. Section 4.1 demonstrated how the inductive bias of ReLU networks produces uncertainty estimates that are not indicative of the familiarity of data to the model; instead, they converge to fix points in the limit. We were able to prove this formally and get an intuition of potential pitfalls in practice in Section 4.1.5, however text classification models in

NLP possess different and more complex architecture, for which similar arguments are not easily applicable. Therefore, we followed up with an empirical investigation into many popular models and uncertainty metrics on three different languages and tasks in Section 4.2. This came with some surprising insights, for instance that uncertainty can be unreliable on OOD data, and that more training or finetuning data can lead to decreased uncertainty quality.

As we argued in the introduction in Section 1.3, data scarcity and the complexity of language are two core features that differentiate uncertainty quantification in NLP from other modalities. In this chapter we discussed the arguably easier setting of text classification: In text classification, we can treat predictions on a sequence-level as i.i.d., and the set of classes is usually much smaller than the number of tokens in a vocabulary. This is why we now turn our attention to the more challenging problem of language generation in the next chapter.

		Task (ID/OOD)		Calibration (ID/OOD)			Uncertainty (ID/OOD)			
	Model	Acc.↑	F_1 ↑	ECE↓	% Cov.↑	⌀Width↓	AUROC↑	AUPR↑	Token τ ↑	Seq. τ ↑
English	LSTM	.79 ±.00	.62 ±.01	.78 ±.00	**1.00** ±.00	144.00 ±.00	.88✚ ±.01	.60✚ ±.01	—	.75○ ±.01
	Bayesian LSTM	.59 ±.06	.46 ±.05	.78 ±.00	.88 ±.00	41.99 ±1.94	.86△ ±.01	.59✖ ±.01	—	.66○ ±.02
	LSTM Ensemble	**.81** ±.00	**.64** ±.00	0.77 ±.00	.87 ±.00	4.27 ±.05	**.92**✚ ±.00	**.71**✚ ±.01	—	.73□ ±.01
	Var. Bert	.45 ±.16	.34 ±.13	.78 ±.00	1.00 ±.00	115.11 ±11.38	.80✖ ±.01	.53✖ ±.01	—	.57○ ±.09
	DDU Bert	.79 ±.00	.64 ±.01	**.77** ±.00	.82 ±.00	**1.46** ±.04	.88○ ±.00	.62○ ±.01	—	**.87**○ ±.00
Danish	LSTM	.93 ±.00 / .92 ±.00	.26 ±.01 / .19 ±.01	.17 ±.00 / .17 ±.00	**1.00** ±.00 / **1.00** ±.00	19.00 ±.00 / 19.00 ±.00	.50○ ±.02	.14○ ±.01	.50○ ±.01 / .47○ ±.00	−.26✚ ±.02 / −.28○ ±.05
	Var. LSTM	.90 ±.02 / .90 ±.02	.08 ±.02 / .09 ±.02	.17 ±.00 / .17 ±.00	.99 ±.01 / .98 ±.01	6.62 ±.37 / 6.68 ±.33	.60✚ ±.04	.21✚ ±.02	.23○ ±.06 / .23○ ±.05	−.04✖ ±.02 / −.02□ ±.05
	ST-τ LSTM	.92 ±.00 / .92 ±.00	.12 ±.00 / .09 ±.00	.17 ±.00 / .17 ±.00	1.00 ±.00 / .99 ±.00	7.10 ±.07 / 7.03 ±.08	.54✚ ±.01	.15✚ ±.01	.50○ ±.00 / .48○ ±.00	−.05□ ±.03 / −.01□ ±.05
	Bayesian LSTM	.93 ±.00 / .93 ±.00	.07 ±.00 / .07 ±.00	.17 ±.00 / .17 ±.00	1.00 ±.00 / 1.00 ±.00	1.68 ±.04 / 1.70 ±.05	.65⬠ ±.17	.31⬠ ±.30	.53○ ±.01 / **.55**○ ±.01	−.01□ ±.07 / −.02✚ ±.04
	LSTM Ensemble	**.95** ±.00 / **.94** ±.00	**.33** ±.01 / **.25** ±.01	0.16 ±.00 / **0.16** ±.00	.98 ±.00 / .97 ±.00	**1.62** ±.00 / **1.58** ±.01	.60□ ±.02	.18□ ±.01	.44□ ±.00 / .45□ ±.00	−.19✚ ±.01 / −.28□ ±.01
	SNGP Bert	.22 ±.35 / .19 ±.34	.03 ±.03 / .02 ±.02	.17 ±.00 / 0.17 ±.00	1.00 ±.00 / 1.00 ±.00	18.84 ±.32 / 18.83 ±.34	.86△ ±.06	.49△ ±.12	.17□ ±.09 / .26□ ±.14	**.29**✖ ±.03 / **.44**□ ±.11
	Var. Bert	.94 ±.00 / .89 ±.00	.29 ±.01 / .17 ±.00	**0.16** ±.00 / 0.16 ±.00	.99 ±.00 / .98 ±.00	2.25 ±.01 / 3.86 ±.08	.86✚ ±.01	.46✚ ±.02	.42○ ±.00 / .17⬠ ±.00	−.35□ ±.01 / −.41□ ±.01
	DDU Bert	.92 ±.00 / .89 ±.00	.25 ±.00 / .17 ±.00	0.16 ±.00 / 0.16 ±.00	.99 ±.00 / .99 ±.00	3.48 ±.01 / 4.04 ±.03	.86○ ±.01	.39○ ±.02	**.56**○ ±.00 / .25○ ±.01	−.24○ ±.01 / −.38○ ±.03
Finnish	LSTM	.75 ±.00 / .69 ±.00	.57 ±.00 / .53 ±.00	.07 ±.00 / .07 ±.00	1.00 ±.00 / 1.00 ±.00	16.00 ±.00 / 16.00 ±.00	.63△ ±.01	.69✚ ±.01	.29○ ±.00 / .19○ ±.01	−.28✚ ±.02 / −.27✚ ±.02
	Var. LSTM	.27 ±.00 / .26 ±.00	.03 ±.00 / .03 ±.00	.07 ±.00 / .07 ±.00	.97 ±.00 / .96 ±.00	1.35 ±.23 / 1.37 ±.21	.51✚ ±.01	.59✚ ±.01	.00△ ±.01 / .00⬠ ±.00	.01△ ±.03 / .01□ ±.01
	ST-τ LSTM	.76 ±.00 / .71 ±.00	.58 ±.00 / .55 ±.00	.06 ±.00 / .06 ±.00	.97 ±.00 / .96 ±.00	3.32 ±.01 / 3.57 ±.01	.62△ ±.01	.69✚ ±.01	.31○ ±.00 / .21○ ±.01	−.14✚ ±.02 / −.12□ ±.04
	Bayesian LSTM	.27 ±.00 / .26 ±.00	.03 ±.00 / .03 ±.00	.07 ±.00 / .07 ±.00	1.00 ±.00 / 1.00 ±.00	16.00 ±.00 / 16.00 ±.00	.51⬠ ±.01	.60✖ ±.00	.00⬠ ±.00 / .00⬠ ±.00	.01○ ±.01 / .04✚ ±.00
	LSTM Ensemble	.81 ±.00 / .75 ±.00	.62 ±.00 / .57 ±.00	.06 ±.00 / .06 ±.00	.99 ±.00 / .98 ±.00	3.46 ±.01 / 3.80 ±.01	**.67**✚ ±.01	**.74**✚ ±.01	.29○ ±.00 / .19○ ±.01	−.28✚ ±.01 / −.31✚ ±.01
	Var. Bert	.87 ±.00 / .81 ±.00	.74 ±.00 / .70 ±.00	.06 ±.00 / .06 ±.00	.99 ±.00 / .99 ±.00	4.68 ±.03 / 5.19 ±.02	.64△ ±.01	.70○ ±.01	.14○ ±.00 / .08✚ ±.00	−.19✖ ±.00 / −.16✖ ±.01
	SNGP Bert	.18 ±.10 / .17 ±.10	.07 ±.02 / .08 ±.02	.07 ±.00 / .07 ±.00	1.00 ±.00 / .99 ±.01	15.00 ±.00 / 15.00 ±.00	.54△ ±.05	.63△ ±.04	.15□ ±.04 / .15□ ±.03	**.12**□ ±.05 / **.14**□ ±.02
	DDU Bert	.87 ±.00 / .81 ±.00	.72 ±.03 / .68 ±.03	**.06** ±.00 / **.06** ±.00	.94 ±.00 / .91 ±.00	**2.16** ±.06 / **2.31** ±.06	.61○ ±.02	.69○ ±.02	**.39**○ ±.04 / **.26**○ ±.03	−.07○ ±.05 / −.16○ ±.04

Table 4.2: Results on the tested datasets. Task performance is measured by macro F_1 and accuracy, calibration by different calibration errors, the coverage percentage the average prediction set width. For every result, and value on the ID and OOD test set is shown. For English, OOD scores are not available since the OOD set does not contain gold labels, and Token τ is missing due to CLINC being a sequence classification task. Uncertainty quality is evaluated using its ability to discriminate between ID and OOD data, quantified by AUROC and AUPR. Furthermore, Kendall's τ is measured between the uncertainty and losses on a sequence- and token-level. Displayed are mean and standard deviation over five random seeds, with bolding and underlining indicating almost stochastic dominance with $\varepsilon_{\min} \leq 0.3$ over all other models. For last section, the best value over uncertainty metrics is given, with symbols indicating the type of metric achieving it: ○ Max. probability, △ Predictive entropy. □ Class variance. ⬠ Softmax gap. ✚ Dempster-Shafer. ✖ Mutual information.



5 Uncertainty in Natural Language Generation

"*Obviously, a computer program that succeeded in generating sentences of a language would be, in itself, of no scientific interest unless it also shed some light on the kinds of structural features that distinguish languages from arbitrary, enumerable sets.*"
—Noam Chomsky in *Formal properties of grammars* (1963).

Natural language generation (NLG) is a multi-faceted field spanning applications such as machine translation (MT), language modeling (LM), summarization, question-answering and dialogue generation. Owing to the recent success of large language models (LLMs) such as GPT-4 (OpenAI, 2023), Bloom (Scao et al., 2022) or Llama (Touvron et al., 2023a), natural language is increasingly used as an interface for end users to interact with models. In order to generate the tokens in a sentence, models typically predict a distribution over subword tokens at every step of the generation process. Due to the paraphrastic nature of language discussed in Section 2.1.3, there is a large uncertainty about which token to select, since there might not be a single "correct" token. Futhermore, just using the most likely token often results in text of low-quality (Holtzman et al., 2020; See et al., 2019; Eikema and Aziz, 2020; Zhang et al., 2021b; Eikema, 2024). For this reason, this uncertain decision is often realized through specialized sampling procedures. However, it has been shown that sampling from the tail of the token distribution also negatively impacts text quality, which is why token distributions are often truncated in practice (Holtzman et al., 2020; Fan et al., 2018; Meister et al., 2023). While this kind of sampling allows for more fluent and varied text, there are no guarantees about the plausibility of the generated text. This is particularly relevant for generation scenarios where pre-trained models are applied to new data whose distribution is different from the training data, increasing the risk of generating erroneous, misleading, and potentially harmful text (Ji et al., 2023b; Guerreiro et al., 2023b; Pan et al., 2023; Alkaissi and

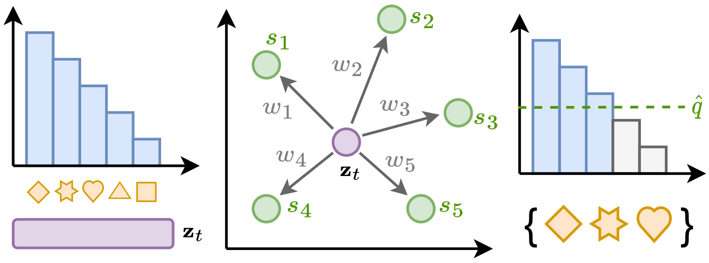


Figure 5.1: Schematic representation of our approach. A decoder hidden representation $\mathbf{z}_t$ is used during inference to retrieve the nearest neighbors and their non-conformity scores s_k. Their relevance is determined by using their distance to compute weights w_k, resulting in the quantile $\hat{q}$ that forms conformal prediction sets.

McFarlane, 2023; Azamfirei et al., 2023). Therefore, this chapter introduces a way of creating calibrated prediction sets to sample from for natural language generation, imbued with the guarantees of conformal prediction.

5.1 Conformalizing Natural Language Generation

The following work is based on Ulmer et al. (2024c).

Conformal prediction (Vovk et al., 2005; Papadopoulos et al., 2002; Angelopoulos and Bates, 2021), has recently gained popularity by providing calibrated prediction sets that are equipped with statistical guarantees about containing the correct solution (see for instance the introduction in Section 2.2.1). Nevertheless, applying conformal prediction to NLG is not trivial: The autoregressive generation process breaks the independence and identical distribution (i.i.d.) assumption underlying conformal prediction techniques, since new predictions are conditioned on the sequence generated so far. We tackle this problem by drawing inspiration from recent advances in nearest-neighbor language modeling (Khandelwal et al., 2020; He et al., 2021a; Xu et al., 2023a) and machine translation (Khandelwal et al., 2021; Zheng et al., 2021; Meng et al., 2022b; Martins et al., 2022). This way, we can dynamically generate calibration sets during inference that maintain statistical guarantees. We schematically illustrate non-exchangeable conformal nucleus sampling in Figure 5.1: In the first step, we obtain a (sorted) probability distribution over tokens and a latent representation $\mathbf{z}_t$ for the current generation step from the model. In a second step, we use the latent representation to query a datastore for similar, previously stored representations and their corresponding non-conformity scores, s_i. In the same way as in the standard conformal prediction algorithm, these non-conformity scores indicate how much a prediction conforms to the rest of the calibration set and its difficulty for the model. These scores are then used to compute a threshold $\hat{q}$ based on the theory of non-exchangeable conformal prediction (Barber et al., 2023), which defines a smaller set of tokens that is sampled from.[55] The extension by Barber et al. allows us to compensate a lack of i.i.d. data by instead defining relevance weights between the test point and the calibration set.

Contributions. We present a general-purpose extension of the conformal framework to NLG by tackling the problems above. Our contributions are as follows: First, to the best of our knowledge, we are the first to present a novel technique based on *non-exchangeable* conformal prediction and to apply it to language generation to produce calibrated prediction sets using a theoretically sound motivation. Secondly, we validate the effectiveness of the method in a language modeling and machine translation context, evaluating the coverage of the calibrated prediction sets and showing that our method is on par with or even outperforms other sampling-based techniques in terms of generation quality, all while maintaining tighter prediction sets and better coverage. Lastly, we demonstrate that these properties are also maintained under distributional shift induced by corrupting the model's latent representations.

5.2 Background

We already discussed the basic formulation of conformal prediction in Section 2.2.1: We first define a non-conformity score that provides an estimate of the distance of the test point to the rest of the data. Then, we determine $\hat{q}$ as the $\lceil (N+1)(1-\alpha)/N \rceil$-th quantile of the non-conformity scores on a held-out set. Finally, we can create calibrated prediction sets of the form

$$\mathcal{C}(\mathbf{x}') = \big\{ y \; \big| \; P_{\boldsymbol{\theta}}(y \mid \mathbf{x}') \geq 1 - \hat{q} \big\}. \tag{5.1}$$

Here, $\mathbf{x}'$ is a new test point for which we could like to construct a prediction set. If a test point $\mathbf{x}'$ and the calibration set are i.i.d., then this set fulfils the conformal guarantee

$$p\big(y' \in \mathcal{C}(\mathbf{x}')\big) \geq 1 - \alpha. \tag{5.2}$$

[55] For simplicity, the figure depicts the simplest form of prediction sets used in conformal prediction. In practice, we use the adaptive prediction sets explained in Section 5.3.

Nevertheless, this formulation is not directly applicable to NLG, as autoregressive generation violates the i.i.d. assumption: If we compare the token distributions at different time steps and different sequences, they will hardly be comparable.

Non-exchangeable Conformal Prediction. Barber et al. (2023) address this shortcoming: When a test point and the calibration data are not i.i.d.,[56] the distributional drift causes any previously found $\hat{q}$ to be miscalibrated, so the intended coverage bound of $1-\alpha$ can no longer be guaranteed. However, we can still perform conformal prediction by assigning a weight $w_i \in [0,1]$ to every calibration data point, reflecting its relevance—i.e. assigning lower weights to points far away from the test distribution. Then, by normalizing the weights with $\tilde{w}_i = w_i/(1+\sum_{i=1}^N w_i)$, we define the quantile as

$$\hat{q} = \inf\big\{q \,\big|\, \sum_{i=1}^{N} \tilde{w}_i \mathbb{1}\big(s_i \leq q\big) \geq 1-\alpha\big\}. \tag{5.3}$$

The construction of the prediction sets then follows the same steps as before. Most notably, the coverage guarantee in Equation (5.2) now changes to

$$p\big(y' \in \mathcal{C}(\mathbf{x}')\big) \geq 1-\alpha-\sum_{i=1}^{N} \tilde{w}_i \varepsilon_i, \tag{5.4}$$

with an extra term including the *total variation distance* (d_{TV}) between the distribution of a calibration and a test point, $\varepsilon_i = d_{\text{TV}}\big((\mathbf{x}_i, y_i), (\mathbf{x}', y')\big)$.[57] Unfortunately, this term is hard to estimate or bound, nevertheless, the selection of appropriate weights that captures the relevance of calibration points to the test set should moderate both the impact of the distant data points on the estimation of the prediction set and the impact of d_{TV} on the coverage bound. In other words, for large d_{TV} values we expect to have smaller weights, that allow us to achieve coverage close to the desired values. We show in our experiments that the loss of coverage when using weights derived from the distance to nearest neighbor is limited, and revisit the practical implications in Section 5.5.

[56] In fact, the coverage guarantee in Equation (5.2) applies to the case where the data is *exchangeable*, a weaker requirement than i.i.d. Specifically, a series of random variables is exchangeable if their joint distribution is unaffected by a change of their order. The work by Barber et al. (2023) allows us to also forgo this requirement.

[57] In this expression, $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}', y')$ denote random variables and the total variation distance is between the two underlying distributions. See Barber et al. (2023) for details.

5.3 Method

We now present a novel method to apply conformal prediction in NLG by synthesizing the non-exchangeable approach of Barber et al. (2023) with k-NN search-augmented neural models (Khandelwal et al., 2021, 2020). In the latter case, the token distribution at the current generation step is interpolated with the predictive distributions of nearest neighbors in a datastore.

A related approach for conformal prediction for NLG by Ravfogel et al. (2023) calibrates prediction sets using the standard conformal procedure described in Section 5.2. In order to improve its effectiveness, the authors also determine multiple $\hat{q}$ values based on the entropy of the token distribution, grouping inputs into one of multiple bins. However, this implies that we would use semantically unrelated (sub-)sequences to calibrate the model—in fact, we show experimentally that this approach generally obtains trivial coverage by producing extremely wide prediction sets. Instead, we propose to perform a *dynamic* calibration step during model inference, only considering the most relevant data points from the calibration set. We do this in the following way: Given a dataset $\{(\mathbf{x}^{(i)}, y^{(i)})\}$ of sequences $\mathbf{x}^{(i)} = (\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_S^{(i)})$ and corresponding references consisting of gold tokens $y^{(i)} = (y_1^{(i)}, \ldots, y_T^{(i)})$, we extract the model's decoder activations $\mathbf{z}_t^{(i)} \in \mathbb{R}^d$ and conformity scores $s_t^{(i)}$.[58] We save those in an optimized datastore, allowing for fast and efficient nearest-neighbor search using the FAISS method by Johnson et al. (2019) through techniques such as quantization and GPU acceleration. In the inference phase, during every decoding step, we then use the decoder hidden state $\mathbf{z}_t'$ to query the data store for the K nearest neighbors and their non-conformity scores and record their distances. We use the squared l_2 distance to compute the weight w_k as

$$w_k = \exp\big(-\big|\big|\,\mathbf{z}_t - \mathbf{z}_k\big|\big|_2^2 \,/\, \tau\big), \tag{5.5}$$

where τ corresponds to a temperature hyperparameter.[59] This formulation is equivalent to a radial basis function kernel with scale parameter τ. Finally, we use the weights to compute the

[58] In this phase, we do not let the model generate freely, but feed it the gold prefix during the decoding process to make sure that conformity scores can be computed correctly.

[59] Using this formulation of the weights w_k that depends on the data deviates from the assumptions of original proof, as discussed in Barber et al. (2023), section 4.5. Nevertheless, our results in Section 5.4 and those by Farinhas et al. (2024) show that the obtained bound in Equation (5.4) still remains useful.

quantile $\hat{q}$ as in Equation (5.3). The entire algorithm is given in Algorithm 2.

Algorithm 2 Non-exchangeable Conformal Language Generation with Nearest Neighbors

```
Require: Sequence $\mathbf{x}$, model $f_{\boldsymbol{\theta}}$, datastore $\text{DS}(\cdot)$ with model activations collected from held-out set, temperature $\tau$

while generating do
    ▷ 1. Extract latent encoding for current input
    $\mathbf{z}_t \leftarrow f_{\boldsymbol{\theta}}(\mathbf{x}_t; y_{<t})$

    ▷ 2. Retrieve $K$ neighbors & non-conformity scores
    $\{(\mathbf{z}_1, s_1), \ldots (\mathbf{z}_K, s_K)\} \leftarrow \text{DS}(\mathbf{z}_t)$

    ▷ 3. Compute weights $w_k$ and normalize
    $w_k \leftarrow \exp(-||\,\mathbf{z}_t - \mathbf{z}_k||_2^2 \,/\, \tau)$
    $\tilde{w}_k \leftarrow w_k/(1 + \sum_{k=1}^{K} w_k)$

    ▷ 4. Find quantile $\hat{q}$
    $\hat{q} \leftarrow \inf\{q \mid \sum_{i=1}^{N} \tilde{w}_i \mathbb{1}\big(s_i \leq q\big) \geq 1 - \alpha\}$

    ▷ 5. Create prediction set
    $\hat{c} \leftarrow \sup\{c' \mid \sum_{j=1}^{c'} P_{\boldsymbol{\theta}}(y = \pi(j) \mid \mathbf{x}_t, y_{<t}) < \hat{q}\} + 1$
    $\mathcal{C}(\mathbf{x}_t) \leftarrow \{\pi(1), \ldots, \pi(\hat{c})\}$

    ▷ 6. Generate next token
    $y_t \leftarrow \text{generate}(\mathcal{C}(\mathbf{x}_t))$
end while
```

Adaptive Prediction Sets. The efficacy of conformal prediction hinges on the choice of non-conformity score, with the simple non-conformity score $s_i = 1 - P_{\boldsymbol{\theta}}(y_t \mid \mathbf{x}, y_{<t})$ known to undercover hard and overcover easy subpopulations of the data (Angelopoulos and Bates, 2021). Due to the diverse nature of language, we therefore opt for *adaptive prediction sets* (Angelopoulos et al., 2021; Romano et al., 2020). Adaptive prediction sets redefine the non-conformity score as the cumulative probability over classes (after sorting in descending order) necessary to reach the correct class. Intuitively, this means that we include all classes whose cumulative probability does not surpass $\hat{q}$. Compared to the simple conformity score, this produces wider predictions sets for hard inputs, encompassing more potentially plausible continuations in a language context. More formally, let π be a permutation function mapping

all possible output tokens $[C]$ to the indices of a permuted version of the set, for which tokens are sorted in descending oder by their probability under the model. We define the non-conformity score as

$$s_i = \sum_{j=1}^{\pi(y_t)} P_{\boldsymbol{\theta}}(\pi^{-1}(j) \mid \mathbf{x}, y_{<t}). \tag{5.6}$$

Since we only include the cumulative mass up until the gold label, the summation stops at $\pi(y)$. The prediction sets are then defined as

$$\mathcal{C}(\mathbf{x}, y_{<t}) = \Big\{\pi^{-1}(1), \ldots, \pi^{-1}(\hat{c})\Big\}, \tag{5.7}$$

with $\hat{c} = \sup\{c' \mid \sum_{j=1}^{c'} P_{\boldsymbol{\theta}}(\pi^{-1}(j) \mid \mathbf{x}, y_{<t}) < \hat{q}\} + 1$, where we add one extra class to avoid empty sets.

5.4 Experiments

In the following sections, we conduct experiments in both language modeling and machine translation. For machine translation we opt for the 400 million and 1.2 billion parameter versions of the M2M100 model (Fan et al., 2021) on the WMT-2022 shared task datasets for German to English and Japanese to English (Bojar et al., 2017). For language modeling, we use the 350 million and 1.3 billion parameter versions of the OPT model (Zhang et al., 2022b) and replicate the setup by Ravfogel et al. (2023): We calibrate our model on 10000 sentences from a 2022 English Wikipedia dump (Wikimedia Foundation, 2022) and test coverage and generation on 1000 sentences from OpenWebText (Gokaslan et al., 2019).[60] All models are used in a zero-shot setup *without extra training or finetuning*. For the datastore, we use the implementation of the FAISS library (Johnson et al., 2019), computing 2048 clusters in total and probing 32 clusters per query. We also summarize the environmental impact of our experiments in Appendix C.2.

5.4.1 Evaluating Coverage

First of all, we demonstrate that the retrieved information from the data store enables us to successfully obtain calibrated prediction

[60] Data obtained through the Hugging Face `datasets` package (Lhoest et al., 2021): `https://huggingface.co/datasets/wikipedia` and `https://huggingface.co/datasets/stas/openwebtext-10k`.

sets. *Coverage* is an important notion in conformal prediction, referring to the correct label being included in a prediction set or interval. Since we can always achieve coverage trivially by choosing the largest possible prediction set, an ideal method strikes a balance between high coverage and small prediction sets. While it is not possible to measure coverage in a free generation setting (see next section), we can assess whether the correct class is contained in the prediction set if we feed the actual reference tokens into the decoder and check whether we include the true continuation.[61] For our MT task, this is reminiscent of an interactive translation prediction setup (Knowles and Koehn, 2016; Peris et al., 2017; Knowles et al., 2019), where we propose possible continuations to a translator, suggesting the next word from a set of words that (a) contains plausible options and (b) is limited in size, in order to restrict the complexity for the end user. Before we run our experiments, we need to determine τ, which we tune on the calibration set using a stochastic hill-climbing procedure described in Appendix C.8. We compare our *non-exchangeable conformal nucleus sampling* (*Non-Ex. CS*) with the following sampling methods: Nucleus sampling (*Nucleus*; Holtzman et al., 2020), which includes all tokens up to a pre-defined cumulative probability mass, and the conformal nucleus sampling (*Conf.*; Ravfogel et al., 2023) discussed earlier. The latter bins predictions on a calibration set by the entropy of the output distribution, and compute one $\hat{q}$ per such entropy bin using the standard conformal procedure given in the beginning of Section 5.2.

Evaluation. We measure the total coverage using different distance metrics, namely, squared l_2 distance, normalized inner product, and cosine similarity (see Tables 5.1 and 5.2),[62] as well as binning predictions by set size and then measuring the per-bin coverage in Figure 5.2 (more results given in Appendix B.7). We also summarize the plots in Figure 5.2 via the *expected coverage gap* (ECG)[63] that we define as

$$\text{ECG} = \sum_{b=1}^{B} \frac{|\mathcal{B}_b|}{N} \max\Big(1 - \alpha - \text{Coverage}\big(\mathcal{B}_b\big), 0\Big), \tag{5.8}$$

[61] We emphasize that access to gold tokens is not required by our method and only done here to measure the actual coverage.

[62] For inner product and cosine similarity, we follow the same form as Equation (5.5), omitting the minus. We normalize the inner product by the square root of the latent dimension.

[63] This is inspired by the expected calibration error (Guo et al., 2017), comparing coverage to $1 - \alpha$, where overcoverage is not penalized due to Equation (5.2)'s lower bound.

where $\mathcal{B}_b$ denotes a single bin and N the total number of considered predictions in the dataset.[64] The ECG thus captures the average weighted amount of undercoverage across bins. In our experiments, we use 75 bins in total. The same bins are used to also evaluate the *size-stratified coverage metric* (SSC) proposed by Angelopoulos et al. (2021), with a well-calibrated method resulting in a SCC close to the desired coverage $1 - \alpha$:

$$\text{SCC} = \min_{b \in \{1,\ldots,B\}} \text{Coverage}\big(\mathcal{B}_b\big). \tag{5.9}$$

We can therefore understand the SCC as the worst-case coverage across all considered bins. We present some additional experiments where we assess the impact of key hyperparameters in Appendix B.8.

			de → en					ja → en				
	Method	Dist.	τ	% Cov.	∅ Width ↓	Scc ↑	Ecg ↓	τ	% Cov.	∅ Width ↓	Scc ↑	Ecg ↓
M2M100$_{(400M)}$	Nucleus	–	–	.9207	.48	.25	.00	–	.9261	.54	.41	.02
	Conf.	–	–	.9951	.94	.33	.03	–	.9950	.96	.14	.00
	Non-Ex. CS	IP	3.93	.8251	.16	.63	.26	11.90	.8815	.24	.67	.03
		l_2	512.14	.8334	.17	.60	.06	419.91	.8468	.18	.61	.05
		cos	2.54	.8371	.17	.63	.06	3.53	.8540	.17	.62	.04
M2M100$_{(1.2B)}$	Nucleus	–	–	.8339	.38	.00	.08	–	.7962	.42	.03	.10
	Conf.	–	–	.9993	.99	.34	.00	–	.9998	.99	.60	.00
	Non-Ex. CS	IP	15.79	.8861	.25	.71	.03	10.45	.9129	.38	.72	.00
		l_2	1123.45	.8874	.25	.72	.03	605.97	.8896	.30	.76	.01
		cos	3.21	.8858	.25	.72	.03	1.48	.8897	.30	.75	.01

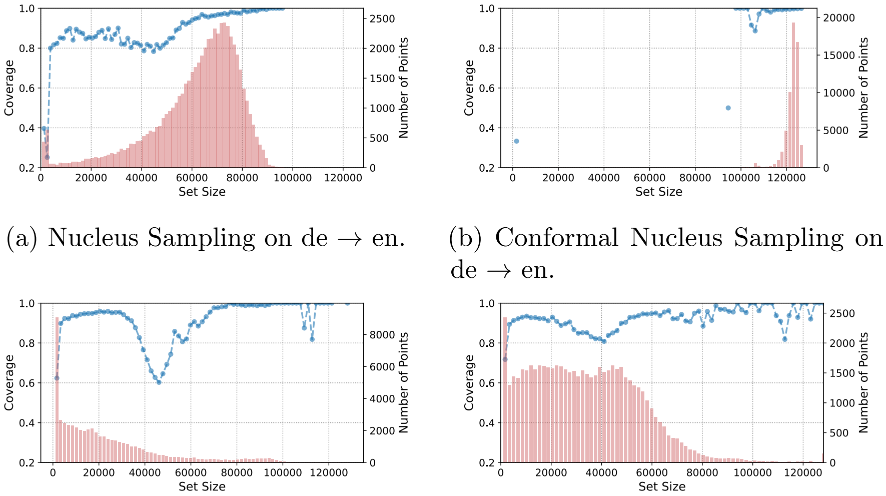
Table 5.1: Coverage results for the de → en and ja → en MT tasks. We report the best found temperature τ while keeping the confidence level α and number of neighbors $k = 100$ fixed. We also show the coverage percentage along with the avg. prediction set size as a proportion of the entire vocabulary (∅ Width) as well as ECG and SSC. Tested distance metrics are inner product (IP), (squared) l_2 distance, and cosine similarity (cos).

Results. The results are shown in Tables 5.1 and 5.2. We found that our method missed the desired coverage of 90% for MT by only 8% or less. Beyond the best values shown in the tables, we were not able to further increase coverage by varying the temperature parameter without avoiding trivial coverage (i.e., defaulting to very large set sizes). This likely due to inherent coverage gap in Equation (5.4) that is due to distributional drift

[64] Since conformal prediction produces a *lower* bound on the coverage, we do not include overcoverage in Equation (5.8).

			OpenWebText				
	Method	Dist.	τ	% Cov.	⌀ Width ↓	Scc ↑	Ecg ↓
$\text{OPT}_{(350\text{M})}$	Nucl. Sampl.	-	-	.8913	.05	.71	.01
	Conf. Sampl.	–	–	.9913	.90	.91	.00
	Non-Ex. CS	IP	4.99	.9352	.19	.80	.00
		l_2	$.31 \times 10^4$	.9425	.17	.80	.00
		cos	4.98	.9370	.15	.83	.00
$\text{OPT}_{(1.3\text{B})}$	Nucl. Sampl.	–	–	.8952	.05	.00	.01
	Conf. Sampl.	–	–	.9905	.88	0.95	.00
	Non-Ex. CS	IP	.48	.9689	.59	.84	.00
		l_2	1.55×10^4	.9539	.20	.83	.00
		cos	.11	.9512	.20	.875	.00

Table 5.2: Coverage results for the LM task. We report the best found temperature τ while keeping the confidence level α and number of neighbors $k = 100$ fixed. We also show the coverage percentage along with the avg. prediction set size as a proportion of the entire vocabulary (⌀ Width) as well as the ECG and SSC metrics. Tested distance metrics are inner product (IP), (squared) l_2 distance and cos. similarity (cos).



(a) Nucleus Sampling on de → en.

(b) Conformal Nucleus Sampling on de → en.

(c) Non-Ex. Conformal Sampling on de → en.

(d) Non-Ex. CS on de → en with $\text{M2M100}_{(1.2\text{B})}$.

Figure 5.2: Conditional coverage for the M2M100 on de → en with the small 418M model (Figures 5.2a to 5.2c) and using the bigger 1.2B model (Figure 5.2d). We aggregate predictions by set size using 75 equally-spaced bins in total. The blue curve shows the conditional coverage per bin, whereas red bars show the number of binned predictions.

and is challenging to estimate directly.

Most notably, our method was able to achieve better SCC scores while maintaining considerably smaller prediction sets than the baselines on average. The reason for this is illustrated in Figure 5.2: while standard nucleus sampling produces some prediction sets that are small, the total coverage seems to mostly be achieved by creating very large prediction sets between 60k–80k tokens. The behavior of conformal nucleus sampling by Ravfogel et al. (2023) is even more extreme in this regard, while our method produces smaller prediction sets, with the frequency of larger set sizes decreasing gracefully. In Figure 5.2d, we can see that the larger M2M100 models also tend to produce larger prediction sets, but still noticeably smaller than the baselines. Importantly, for both M2M100 models, even very small prediction sets (size ≤ 1000) achieve non-trivial coverage, unlike the baseline methods.

For LM, we always found the model to slightly *over*cover. This does not contradict the desired lower bound on the coverage in Equation (5.4) and suggests a more negligible distributional drift. While nucleus sampling produces the smallest average prediction sets, we can see that based on the SCC values some strata remain undercovered. Instead, our method is able to strike a balance between stratified coverage and prediction set size. With respect to distance measures, we find that the difference between them is minimal, indicating that the quality largely depends on the retrieved local neighborhood of the decoder encoding and that finding the right temperature can help to tune the models to approximate the desired coverage. We would now like to find out whether this neighborhood retrieval mechanism can prove to be robust under distributional shift as well. Since we did not observe notable differences between the distance metrics, we continue with the l_2 distance.

5.4.2 Coverage Under Shift

To demonstrate how the retrieval of nearest neighbors can help to maintain coverage under distributional shift, we add Gaussian noise of increasing variance—and therefore intensity—to the last decoder hidden embeddings (for MT) and the input embeddings (LM).[65] This way, we are able to simulate distributional drift while still keeping the original sequence of input tokens intact, allowing us to measure the actual coverage. We show the achieved coverage along with the average set size (as a percentage of the

[65] A similar approach can be found for instance in the work of Hahn and Choi (2019); Zhang et al. (2023c) or by Snoek et al. (2019); Hendrycks and Dietterich (2019) in a computer vision context.

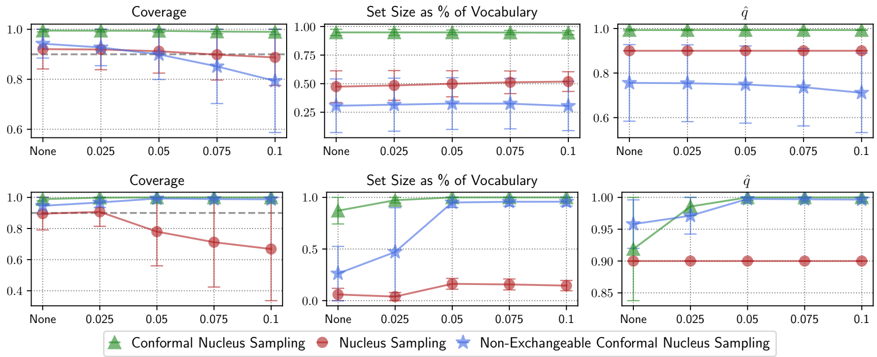


Figure 5.3: Coverage, average set size and $\hat{q}$ based on the noise level on the de → en MT task (top) and open text generation task (bottom). Error bars show one standard deviation.

	Noise level				
	None	.025	.05	.075	.1
⌀ Entropy	8.46	8.71	9.20	9.71	10.08
Nucl. Sampl. (ρ)	.87	.86	.84	.82	.81
Conf. Sampl. (ρ)	.60	.60	.60	.57	.55
Non-Ex. CS (ρ)	−.14	−.18	−.27	−.37	−.45

Table 5.3: Average entropy of 400M M2M100 model on de → en per noise level as well as the Spearman's ρ correlation coefficients between the predictive entropy and the prediction set size of the different methods. All results are significant with $p < 0.0001$.

total vocabulary) and the average quantile $\hat{q}$ in Figure 5.3. We can see that the conformal sampling method deteriorates into returning the full vocabulary as a prediction set. Thus it behaves similarly to simple sampling as indicated by the $\hat{q}$ values being close to 1. Nucleus sampling provides smaller prediction sets compared to conformal sampling, but they seem invariant to noise. As such, the method is not robust to noise injection in the open text generation task, and the obtained coverage deteriorates with noise variance ≥ 0.025. Instead, the use of nearest neighbors allows for the estimation of prediction sets that are small but amenable to increase, such that the obtained coverage remains close to the desired one. We can specifically observe that the prediction set size increases considerably to mitigate the injected noise in the open-text generation case.

Neighbor Retrieval. We further analyze how the retrieval enables this flexibility by relating it to the entropy of the output distribution of the 400M parameters M2M100 on German to English. Intuitively, the baseline methods, faced by high-entropy output distributions, need to produce wide prediction sets in order to maintain coverage. In fact, we report such results by correlating entropy levels and prediction set sizes using Spearman's ρ in Table 5.3, showing strong positive correlations. Our method in contrast consistently shows an *anti*correlation between these two quantities, enabled by decoupling the creation of prediction sets from statistics of the output distribution to instead considering the non-conformity scores of similar subsequences. The fact that the prediction set size is not just dependent on the entropy of the predictions while maintaining coverage demonstrates the value of the nearest neighbors: In this way, model uncertainty becomes more flexible and is corroborated by evidence gained from similar inputs.

5.4.3 Generation Quality

Crucially, our method should not degrade and potentially even improve generation quality. Thus, we evaluate the generation quality for the same tasks without supplying the gold prefix, instead employing standard language generation procedures. For language modeling, we follow Ravfogel et al. (2023) and use the first 35 tokens from the original sentence as input. We compare against a set of generation strategies including top-k sampling (Fan et al., 2018; Holtzman et al., 2018; Radford et al., 2019), nucleus sampling and conformal nucleus sampling. We also test a variant of our method using constant weights $w_k = 1$ for retrieved neighbors (*Const. Weight CS*) to assess the impact of the weighted neighbor retrieval procedure. We further compare with beam search (Medress et al., 1977; Graves, 2012) with a softmax temperature of 0.1, and greedy decoding. Evaluation is performed using BLEU (Papineni et al., 2002), COMET-22 (Rei et al., 2020, 2022) and chrF (Popović, 2017) for MT, where COMET-22 is a trained neural metric. For text generation, we use MAUVE (Pillutla et al., 2021) and BERTscore (Zhang et al., 2020c).[66] MAUVE is a neural metric that measures the divergence from human-written text, while BERTscore involves a fine-tuned Bert model that aims to predict human quality judgments.

[66] All metrics except for COMET were used through Hugging Face `evaluate`. MAUVE uses `gpt2` as a featurizer.

		de → en			ja → en		
	Method	BLEU ↑	COMET ↑	CHRF ↑	BLEU ↑	COMET ↑	CHRF ↑
$\text{M2M100}_{(400\text{m})}$	Beam search	28.53	0.88	55.58	11.37	.63	37.74
	Greedy	27.81	.90	54.9	10.73	.58	36.5
	Nucleus Sampling	27.63 ±.03	.89 ±.01	54.80 ±.07	10.61 ±.15	.59 ±.01	36.52 ±.19
	Top-k Sampling	27.63 ±.03	.89 ±.01	54.79 ±.07	10.61 ±.15	.59 ±.01	36.52 ±.19
	Conf. Sampling	27.63 ±.03	.89 ±.01	54.80 ±.07	10.61 ±.15	.59 ±.01	36.52 ±.19
	Const. Weight CS	27.63 ±.03	.89 ±.01	54.80 ±.07	10.61 ±.15	0.59 ±.01	36.52 ±.19
	Non-Ex. CS	27.65 ±.10	.90 ±.01	54.82 ±.14	$\underline{10.74}$ ±.11	.59 ±.01	36.61 ±.08
$\text{M2M100}_{(1.2\text{B})}$	Beam search	30.89	.90	56.8	13.76	.63	40.43
	Greedy	29.52	.90	55.67	12.94	.60	39.91
	Nucleus Sampling	29.37 ±.12	.90 ±.00	55.55 ±.11	10.61 ±.15	.59 ±.01	36.52 ±.19
	Top-k Sampling	29.53 ±.00	.90 ±.00	55.67 ±.00	12.91 ±.08	.60 ±.01	39.95 ±.00
	Conf. Sampling	29.37 ±.12	.90 ±.00	55.55 ±.11	12.91 ±.08	.60 ±.00	39.95 ±.08
	Const. Weight CS	29.37 ±0.12	.90 ±.00	55.55 ±.11	12.91 ±.08	.60 ±.01	39.95 ±.08
	Non-Ex. CS	29.37 ±0.12	.90 ±.00	55.55 ±.11	12.91 ±.08	.60 ±.01	39.95 ±.08

Table 5.4: Generation results for the de → en and ja → en translation tasks. We report performance using 5 beams for beam-search, top-k sampling with $k = 10$, and nucleus sampling with $p = 0.9$. Conformal methods all use $\alpha = 0.1$, with non-exchangeable variants retrieving 100 neighbors, and sampling uses a softmax temperature of 0.1. Results using 5 different seeds that are stat. significant according to the ASO test (del Barrio et al., 2018a; Dror et al., 2019; Ulmer et al., 2022c) with a confidence level of 0.95 and threshold $\varepsilon_{\min} \leq 0.3$ are underlined.

Results. We show the results for the different methods in Tables 5.4 and 5.5. We see that beam search outperforms all sampling methods for MT. This corroborates previous work by Shaham and Levy (2022) who argue that (nucleus) sampling methods, by pruning only the bottom percentile of the token distribution, introduce some degree of randomness that is beneficial for open text generation but may be less optimal for conditional language generation, where the desired output is constrained and exact matching generations are preferred (which is the case for MT). Among sampling methods, we find nucleus sampling and conformal sampling to perform similarly (being in agreement with the findings of Ravfogel et al., 2023) but are sometimes on par or even outperformed by our non-exchangeable conformal sampling for MT. For text generation, our method performs best for the smaller OPT model, but is slightly beaten by conformal nucleus sampling in terms of MAUVE. When using constant weights, performance deteriorates to the conformal

		OpenWebText	
	Method	MAUVE ↑	BERTScore F_1 ↑
$\text{OPT}_{(350\text{M})}$	Beam search	.12	.79
	Greedy	.02	.79
	Nucleus Sampling	.91 ±.02	.80 ±.00
	Top-k Sampling	.90 ±.03	$\underline{.80}$ ±.00
	Conf. Sampling	.91 ±.02	.80 ±.00
	Const. Weight CS	.91 ±.02	.80 ±.00
	Non-Ex. CS	.92 ±.01	.80 ±.00
$\text{OPT}_{(1.3\text{B})}$	Beam search	.17	.80
	Greedy	.05	.79
	Nucleus Sampling	.91 ±.02	.80 ±.00
	Top-k Sampling	.93 ±.01	$\underline{.81}$ ±.00
	Conf. Sampling	.93 ±.01	.80 ±.00
	Const. Weight CS	.91 ±.02	.80 ±.00
	Non-Ex. CS	.92 ±.01	.81 ±.00

Table 5.5: Generation results for the open text generation. We report performance using 5 beams for beam-search, top-k sampling with $k = 10$, and nucleus sampling with $p = 0.9$. Conformal methods all use $\alpha = 0.1$, with non-exchangeable variants retrieving 100 neighbors. Results using 5 different seeds that are stat. significant according to the ASO test (del Barrio et al., 2018a; Dror et al., 2019; Ulmer et al., 2022c) with a confidence level of 0.95 and threshold $\varepsilon_{\text{min}} \leq 0.3$ are underlined.

sampling setup, emphasizing the importance of not considering all conformity scores equally when computing $\hat{q}$, even though the effect seems to be less pronounced for larger models. This illustrates the benefit of creating flexible prediction sets that are adapted on token-basis, suggesting that both the latent space neighborhoods as well as the conformity scores are informative.

5.5 Discussion

Our experiments have shown that despite the absence of i.i.d. data in NLG and the loss in coverage induced by using dynamic calibration sets, the resulting coverage is still close to the pre-specified desired level for both LM and MT. Additionally, even though the coverage gap predicted by the method of Barber et al. (2023) is infeasible to compute for us, we did not observe any critical degradation in practice. Further, we demonstrated how sampling from these calibrated prediction sets performs similarly or better than other sampling methods. Even though our method is still outperformed by beam search in the MT setting, previous work such as minimum Bayes risk decoding has shown how multiple samples can be re-ranked to produce better outputs (Kumar and Byrne, 2002, 2004; Eikema and Aziz, 2020; Fernandes et al., 2022; Freitag et al., 2023). Additionally, recent dialogue systems based on LLMs use sampling instead of beam search for generation (e.g. OpenAI, 2023; AI@Meta, 2024). Since our prediction sets are more flexible and generally tighter, our results serve as a starting point for future work. For instance, our technique could be used with non-conformity scores that do not consider token probabilities alone (e.g. Meister et al., 2023) or using prediction set widths as a proxy for uncertainty (Angelopoulos et al., 2021). Furthermore, the extension with conformal risk control (Angelopoulos et al., 2023; Farinhas et al., 2024) enables guarantees with respect to a wider family of function than just coverage. This opens up other directions, for instance defining functions that assess the desirability of the current generation, analogous to on-the-fly alignment procedures (Yang and Klein, 2021; Qin et al., 2022; Mudgal et al., 2023; Gao et al., 2024a).

Limitations. We highlight two main limitations of our work here: Potential issues arising from different kinds of dataset shift as well as efficiency concerns. Even though any loss of coverage due to the term quantifying distributional drift in Equation (5.4) was limited in our experiments (see Sections 5.4.1 and 5.4.2), this might not hold across all possible setups. As long as we cannot feasibly approximate the shift penalty, it is impossible to determine a priori whether the loss of coverage might prove to be detrimental, and would have to be checked in a similar way as in our experiments. Furthermore, we only consider shifts between the models' training distributions and test data distributions here, while many other, unconsidered kinds of shifts exist (Moreno-Torres et al., 2012; Hupkes et al., 2023). Additionally, even using optimized tools such as FAISS (Johnson et al., 2019), moving the conformal prediction calibration step to inference incurs additional computational cost during generation. Nevertheless, works such as ?Martins et al. (2022) show that there are several ways to improve the efficiency of k-NN approaches, and we leave such explorations to future work.

5.6 Summary

In this chapter, we successfully demonstrated the application of a non-exchangeable variant of conformal prediction to machine translation and language modeling with the help of k-NN retrieval. By retrieving a calibration set on the fly, one can create prediction sets for language generation based on the non-exchangeable conformal prediction algorithm by (Barber et al., 2023). We demonstrated that this method best maintains the desired coverage across different dataset strata while keeping prediction sets smaller than other sampling methods, all while providing theoretical coverage guarantees about coverage that other comparable methods lack.

However, this method has multiple shortcomings: Except through the width of prediction sets, it does not explicitly quantify the uncertainty of the model, adds computational overhead to the inference process and furthermore requires access to the internal states of the model. This becomes problematic when trying to apply to larger models than for instance M2M100$_{(1.2B)}$: Many of the contemporary open-source models (like those in the case study in Section 3.2.3) comprise 7 billion, 40 billion or even more parameters. In addition, commercial closed-source models that can only be accessed through an API are estimated to be even larger.[67] The nature of the API-only access further exacerbated this problem, as no information internal to the model, sometimes not even the token distribution, can be accessed. The next chapter therefore proposes a method that operates within this very challenging and restrictive setup.

[67] For example, GPT-4's parameter count is rumored to be 1.76 trillion (The Decoder, 2023).

6 | Uncertainty in Large Language Models

> “*In desperation I asked Fermi whether he was not impressed [. . .]. He replied “How many arbitrary parameters did you use for your calculations?” I [. . .] said “Four.” He said: “I remember my friend Johnny von Neumann used to say, with four parameters I can fit an elephant, and with five I can make him wiggle his trunk.” With that, the conversation was over.*”
>
> —Freeman Dyson in *A Meeting with Enrico Fermi* (2006).

The following work is based on Ulmer et al. (2024a).

When given a case description of “*A man superglued his face to a piano and he says it’s making it hard to get a full night of sleep*”, a medical LLM was found to list a plethora of potential causes in its diagnosis, including narcolepsy, sleep apnea and others.[68] This, of course, ignores the seemingly obvious reason for the patient’s complaints. While humorous, this example illustrates the pitfalls of practical LLM applications: Despite often looking convincing on the surface—especially to non-experts—model responses can be wrong or unreliable, leading to potentially harmful outcomes or a loss of trust in the system, foregoing its benefits. Indeed, consistent behavior (imagine e.g. reliably indicating a lack of confidence for unsure responses) has been argued as one way to build trust in automated systems (Jacovi et al., 2021), while misleading predictions have been empirically shown to lead to a loss of trust that can be hard to recover from (Dhuliawala et al., 2023).

The introductory example shows that as large language models (LLMs) are increasingly deployed in user-facing applications, building trust and maintaining safety by accurately quantifying a model’s confidence in its prediction becomes even

[68] https://x.com/spiantado/status/1620459270180569090 (last accessed Nov. 7, 2023).

more important. However, finding effective ways to calibrate LLMs—especially when the only interface to the models is their generated text—remains a challenge. Most previously discussed methods to calibrate model predictions, such as the ones in Section 2.2.1 or even the non-exchangeable conformal language generation from the previous Chapter 5, require some degree of retraining or at least access to model hidden states and / or logits.

In this chapter, we introduce APRICOT 🍑 (**a**uxiliary **pr**ed**i**ction of **co**nfidence **t**argets): A method to set targets to calibrate confidence scores to and train an additional model that predicts an LLM's confidence based on its textual input and output alone. This approach has several advantages: It is conceptually simple, does not require access to the target model beyond its output, does not interfere with the language generation, and has a multitude of potential usages, for instance by verbalizing the predicted confidence or adjusting the given answer based on the confidence. We show how our approach performs competitively in terms of calibration error for white-box and black-box LLMs on closed-book question-answering to detect incorrect LLM answers. Our contributions are as follows: We propose to obtain calibration targets for LLM confidence scores without requiring any additional information about LLM internals or question metadata. We show that using auxiliary models on the target LLM's input and output is sufficient to predict a useful notion of confidence for question-answering on TriviaQA (Joshi et al., 2017) and CoQA (Reddy et al., 2019). We also perform additional studies to identify which parts of the LLM's output are most useful to predict confidence.

6.1 Calibrating LLMs with Auxiliary Models

Method	Black-box LLM?	Consistent?	Calibrated?
Sequence likelihoods	✗	✔	✗
Verbalized uncertainty	✔	✗	✗
APRICOT 🍑 (ours)	✔	✔	✔

Table 6.1: Comparison of appealing attributes that LLM confidence quantification techniques should fulfill. They should ideally be applicable to black-box LLMs, be consistent (i.e. always elicit a response that indicates confidence in contrast to an unrelated response), and produce calibrated estimates of confidence.

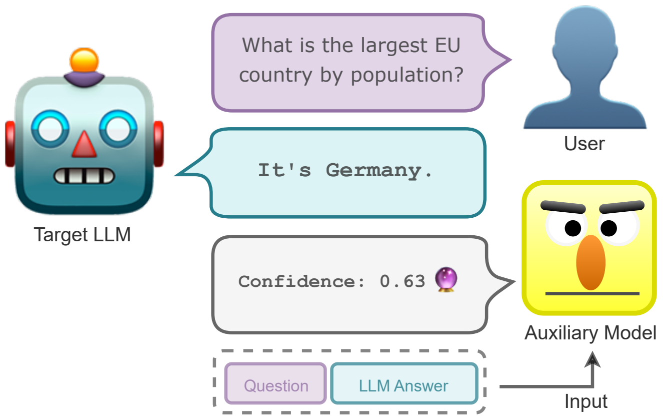


Figure 6.1: Illustration of APRICOT 🍑: We train an auxiliary model to predict a target LLM's confidence based on its input and the generated answer.

Estimating the confidence of an LLM can be challenging, since their size rules out many traditional techniques that require finetuning or access to model parameters. In this light, using the likelihood of the generated sequence might seem like an appealing alternative; however, it may not actually reflect the reliability of the model's answer and often cannot be retrieved when using black-box models, where the only output is the generated text. Verbalized uncertainty, i.e. prompting the LLM to express its uncertainty in words, can be a solution when the model is powerful enough. But as we later show in Section 5.4, the generated confidence expressions are not very diverse, and results are not always *consistent*, meaning that the model does not always generate a desired confidence self-assessment. We will later see how for verbalized uncertainty for instance, models sometimes respond with unrelated answers, even when prompted to express their uncertainty. As we illustrate in Table 6.1, our method, APRICOT 🍑, fulfills all of these criteria: Through a one-time finetuning procedure of an auxiliary model on the target LLMs outputs, we have full control over a calibrated model that gives consistent and precise confidence estimates.

In Figure 6.2 we give an overview of APRICOT 🍑, which consists of three main steps: Firstly, we prompt the target LLM to generate training data for our auxiliary model (Section 6.1.1). Secondly, we set calibration targets in a way that does not require access to the target LLM beyond its generated outputs (Section 6.1.2). Lastly, we train the auxiliary calibrator to predict the target LLM's

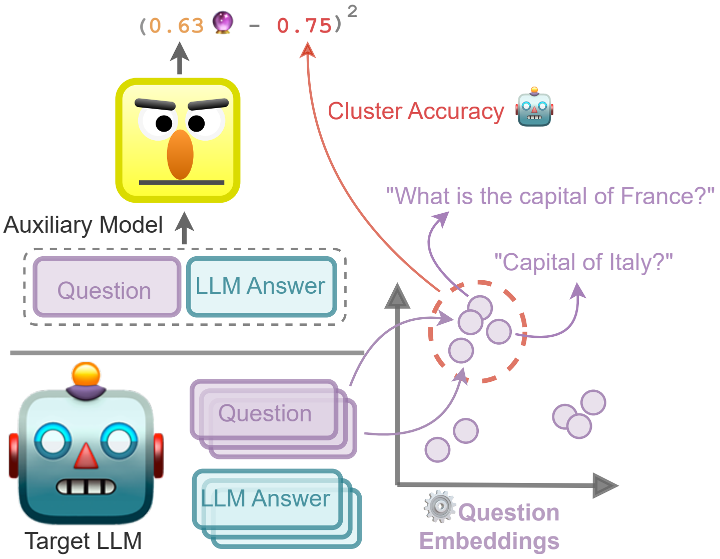


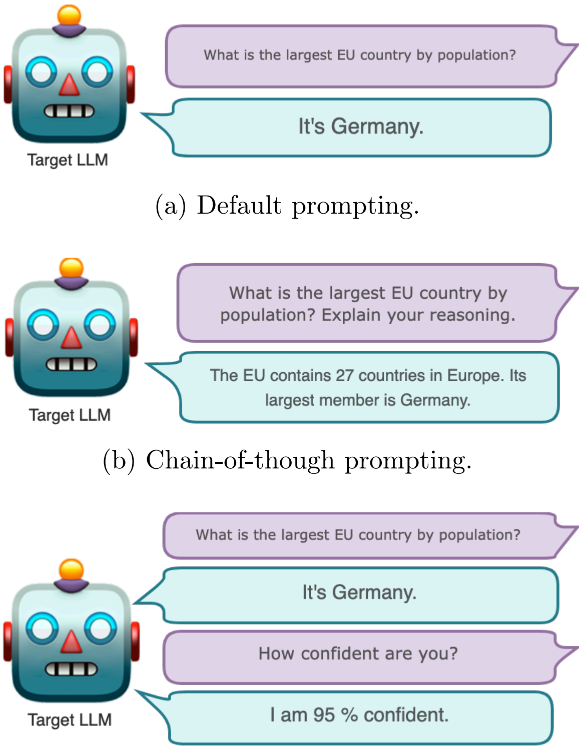
Figure 6.2: Full overview over APRICOT 🍑. We collect a LLM's answer to a set of questions and embed the latter using an embedding model. After clustering similar questions and identifying the LLM's accuracy on them, we can use this value as reference when training to predict the confidence from a question-answer pair.

confidence for a given question (Section 6.1.3).[69] Thereby, we add two parts that are agnostic to the LLM in question: A method that determines calibration targets, and their prediction through the auxiliary model. Note that we use the terms auxiliary model or calibrator interchangeably in the following sections.

6.1.1 Prompting the Target LLM

In the first step, we generate finetuning data for the auxiliary model by prompting the target LLM on the given task. Here, we explore different variations to see which model response might provide the best training signal for the auxiliary calibrator. More concretely, while the original prompt and model generation might already suffice to predict the model's confidence, we also ask the model to elaborate on its answer using chain-of-thought prompting (Wei et al., 2022). We hypothesize that including additional reasoning

[69] In general, using secondary neural models to predict properties of the generated text also has connections to other tasks such as translation quality estimation (Blatz et al., 2004; Quirk, 2004; Wang et al., 2019; Glushkova et al., 2021; Zerva et al., 2022), toxicity classification (Maslej-Krešňáková et al., 2020) or fine-grained reward modeling (Wu et al., 2023b).


What is the largest EU country by population?
It's Germany.
Target LLM
What is the largest EU country by population? Explain your reasoning.
The EU contains 27 countries in Europe. Its largest member is Germany.
Target LLM
What is the largest EU country by population?
It's Germany.
How confident are you?
I am 95 % confident.
Target LLM

(a) Default prompting.

(b) Chain-of-though prompting.

(c) Prompting with verbalized confidence.

Figure 6.3: Illustration of the prompting strategies used to generate the input data for the auxiliary calibrator. Note that (c) can also involve confidence expressed in words ("My confidence level is low") and that (b) and (c) can be combined.

steps exposes signals that are useful for the calibrator.[70] We furthermore take a model's assessment of its confidence into account, too. Recent works on *verbalized uncertainty* (Lin et al., 2022a; Tian et al., 2023) investigated how to elicit such an assessment as a percentage value, e.g. "I am 95 % confident in my answer", or using linguistic expressions such as "My confidence is somewhat low". While previous studies like Zhou et al. (2024) have demonstrated the difficulty in obtaining reliable self-assessments, we can just treat verbalized uncertainties as additional input features, and let their importance be determined through the auxiliary model training. We illustrate the different prompting strategies in Figure 6.3 and elaborate on the prompts in the following.

[70] We do this while acknowledging evidence by Turpin et al. (2023) that shows that any chain-of-thought reasoning might not reflect the actual reasons for a specific model response. Nevertheless, even if chain-of-thought reasoning *does not* unveil the actual process of the LLM, it can provide useful textual features to the auxiliary model, including unexpected intermediate result or linguistic markers of uncertainty.

Prompt Design. We use a simple prompt for question-answering, where we fill in a template of the form “Question: {Question} Answer:”. For in-context samples, we prepend the demonstrations to the input, using the sample template as above. In the case of chain-of-thought prompting, we use the prompting below:

QA Chain-of-thought prompt

> Briefly answer the following question by thinking step by step.
> Question: {Question} Answer:

In the case where the question is supposed to be answered given some context, we slightly change the prompt design:

Chain-of-thought prompt with context

> Context: {Context}
> Instruction: Briefly answer the following question by thinking step by step.
> Question: {Question}
> Answer:

Here, the passage that questions are based on is given first, and chain-of-thought prompting is signaled through the “Instruction” field. When no chain-of-thought prompting is used, this field is omitted. For the verbalized uncertainty, we use the following prompts, in which case we omit any in-context samples:

Verbalized uncertainty prompt (quantitative)

> {Question} {Model answer} Please provide your confidence in the answer only as one of ’Very Low’ / ’Low’ / ’Somewhat Low’ / ’Medium’ / ’Somewhat High’ / ’High’ / ’Very High’:

Verbalized uncertainty prompt (qualitative)

> {Question} {Model answer} Please provide your confidence in the answer only in percent (0–100 %):

We follow Kuhn et al. (2023) and use 10 in-context samples for the original answer, which are randomly sampled from the training set (but in contrast to Kuhn et al., we sample different examples for each instance). When prompting for verbalized uncertainty, we remove these in-context samples. Additionally, verbalized uncertainty expressions such as ‘very low’ or ‘high’ are mapped back onto the following numerical values for evaluation purposes (in the order of appearance in the template above): $0, 0.3, 0.45, 0.5, 0.65, 0.7, 1$.[71]

6.1.2 Setting Calibration Targets

After explaining the inputs to the auxiliary model, the question naturally arises about what the calibrator should be trained to predict. The work by Mielke et al. (2022) introduces an additional model that simply predicts the correctness of an individual answer (and does so by using the target model’s internal hidden states, which is not possible for black-box models). We test this type of output in Section 6.2.2, but we also show that we can produce better calibration targets through clustering.

Recall the notion of calibration and calibration error from Section 2.2.1, where we saw that the expected calibration error can be approximated by binning points into buckets $\mathcal{B}_m$ (Naeini et al., 2015) by confidence:

$$\sum_{m=1}^{M} \frac{|\mathcal{B}_m|}{N} \Bigg| \underbrace{\frac{1}{|\mathcal{B}_m|} \sum_{i \in \mathcal{B}_m} \mathbb{1}\big(\hat{y}_i = y_i\big)}_{\text{Bin accuracy (target)}} - \underbrace{\frac{1}{|\mathcal{B}_m|} \sum_{i \in \mathcal{B}_m} \hat{p}_i}_{\text{Avg. bin confidence}} \Bigg|, \tag{6.1}$$

where $\hat{p}_i$ corresponds to some confidence score. Our key insight is here that we can optimize the expected calibration error through a similar approximation as in Equation (6.1), without changing the LLM’s original answers or access to token probabilities. We can abstract the idea in Equation (6.1) as aggregating samples in homogeneous groups (in the above case, groups of similar confidence), and measuring the group-wise accuracy. But now, instead of creating bins $\mathcal{B}_m$ by confidence, which is not possible in a black-box setting, we create clustered sets $\mathcal{C}_m$ of questions with similar sentence embeddings. Calibration targets are then obtained by using the average accuracy of the LLMs answers per question set $\mathcal{C}_m$. This is similar to the method of Lin et al. (2022a), who consider the accuracy per question category (e.g. multiplication or addition math questions). Yet in the absence of such additional

[71] The choice of these specific numbers is admittedly somewhat arbitrary, and other works such as Lin et al. (2022a) have also employed similarly heuristic scales. Tian et al. (2023) motivate their mapping from expression to probabilities to a social media survey by Fagen-Ulmschneider (2015), where respondents were asked to assign probabilities to different expressions. However, these values vary greatly between participants, and thus assigning a single numerical value is still challenging.

categorization data, we expect good embedding and clustering algorithms to roughly group inputs by category. Höltgen and Williamson (2023) also echo a similar idea of more generalized grouping choices, describing how ECE's grouping by confidence can be abstracted to other kinds of similarities. They also provide a proof that the calibration error of a predictor based on a k-nearest neighbor clustering tends to zero in the infinite data limit.

Implementation. Practically, we embed questions into a latent space using a light-weight model such as SentenceBert (Reimers and Gurevych, 2019), normalize the embeddings along the feature dimension (Timkey and van Schijndel, 2021), and then use HDBSCAN (Campello et al., 2013), an unsupervised, bottom-up clustering algorithm, to cluster them into questions of similar topic. The use of HDBSCAN has multiple advantages: Compared to e.g. k-means, we do not have to determine the numbers of clusters in advance, and since the clustering is conducted bottom-up, clusters are not constrained to a spherical shape. Furthermore, compared to its predecessor DBSCAN (Ester et al., 1996), HDBSCAN does not require one to determine the minimum distance between points for clustering manually. We evaluate this procedure in Section 6.2.1 and Appendix B.9.

6.1.3 Training the Auxiliary Model

After determining the input and the training targets for the auxiliary model in the previous sections, we can now describe the actual training procedure that makes it predict the target LLM's confidence. To start, we feed the questions alongside some in-context samples into our target LLM. We retain the generated answers and create a dataset that combines the question (without in-context samples) and the target model's answers. These are used to train the auxiliary calibrator to predict the calibration targets obtained by the clustering procedure above. In our experiments, we use DeBERTaV3 (He et al., 2023b), an improvement on the original DeBERTa model (He et al., 2021b) using variety of improvements with respect to its architecture and pre-trainign objective. We then finetune it using the AdamW optimizer (Loshchilov and Hutter, 2018) in combination with a cosine learning rate schedule. We minimize the following mean squared error, where $\hat{p}_i$ is the predicted confidence, $\mathcal{C}(i)$ the cluster that the input question with index i belongs to, and $\hat{a}_j$ an answer given by the target LLM:

$$\mathcal{L}\big(\hat{p}_i, \mathcal{C}(i)\big) = \Big(\hat{p}_i - \underbrace{\frac{1}{|\mathcal{C}(i)|}\sum_{j\in\mathcal{C}(i)} \mathbb{1}\big(\hat{a}_j \text{ is correct}\big)}_{\text{Cluster accuracy (target)}}\Big)^2. \tag{6.2}$$

We also explore a variant that simply predicts whether the LLM's answer is expected to be correct or incorrect, so in this case, we simply optimize a binary cross-entropy loss:

$$\mathcal{L}\big(\hat{p}_i, \hat{a}_i\big) = \mathbb{1}\big(\hat{a}_i \text{ is correct}\big)\log \hat{p}_i + \big(1 - \mathbb{1}\big(\hat{a}_i \text{ is correct}\big)\big)\log(1-\hat{p}_i). \tag{6.3}$$

Although omitted here for clarity, the actual loss also uses loss weights to balance the unequal distribution of correct and incorrect language model answers.[72] Finally, we select the final model via the best loss on the validation set. We determine the learning rate and weight decay term through Bayesian hyperparameter search (Snoek et al., 2012), picking the best configuration by validation loss. We detail search ranges and found values in Appendix C.4.4. Training hardware and the environmental impact are discussed in Appendix C.2.

6.2 Experiments

We now demonstrate how APRICOT 🍑 provides a simple yet effective solution to calibrate LLMs. Before assessing the quality of the unsupervised clustering to determine calibration targets from Section 6.1.2, we first introduce the dataset and models.

Datasets. We employ TriviaQA (Joshi et al., 2017), a common (closed-book) question-answering dataset. Open-ended question-answering is an ideal testbed for natural language generation tasks, since it is comparatively easy to check whether an answer is correct or not, so calibration has an intuitive interpretation. To preprocess TriviaQA, we create a training set of 12k examples and choose another 1.5k samples as a validation and test split, respectively.[73] Secondly, we run experiments on CoQA (Reddy et al., 2019), a conversational question-answering dataset in which the model is quizzed about the information in a passage of text. We treat the

[72] The loss weights are based on `scikit-learn`'s implementation using the "balanced" mode, see `https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html`.

[73] Since the original test split does not include answers, we generate the validation and test split from the original validation split.

dataset as an open-book dataset, where the model is shown the passage and then asked one of the corresponding questions at a time. We extract a subset of the dataset to match the split sizes of TriviaQA.

Models. We test two models settings: A white-box setting, where the model can be run locally and we have full access to its internals, and a black-box setting, where the model is only available through an API, drastically reducing the options for uncertainty quantification methods. For our white-box model experiments, we choose a 7 billion parameter variant of the Vicuna v1.5 model (Zheng et al., 2023),[74] an instruction-finetuned model originating from Llama 2 (Touvron et al., 2023a). For the black-box model, we opt for OpenAI's GPT-3.5 (OpenAI, 2022).[75] Despite recent API changes granting access to token probabilities,[76] creating methods for black-box confidence estimation is still relevant for multiple reasons: Token probabilities are not available for most black-box models, they might be removed again to defend against potential security issues; and they are not always a reliable proxy for confidence.

6.2.1 Setting Calibration Targets by Clustering

Before beginning our main experiments, we would like to verify that our proposed methodology in Section 6.1.2 is sound. In particular, clustering the embeddings of questions and computing the calibration confidence targets rests on the assumption that similarly-themed questions are collected in the same cluster. Ideally, we would like to check this using metadata, which however is usually not available.

Setup. Instead, we evaluate this through different means: We first use the `all-mpnet-base-v2` model from the sentence transformers package (Reimers and Gurevych, 2019) and HDBSCAN with a minimum cluster size of 3 to cluster questions. We then analyze the textual and semantic similarity of questions in a cluster by computing the average pair-wise ROUGE-L score (*semantic*; Lin, 2004)[77] between questions, and cosine similarities between question *embeddings* of the same cluster (*semantic*). Since we assume the

[74] https://huggingface.co/lmsys/vicuna-7b-v1.5.

[75] Specifically, using version `gpt-3.5-turbo-0125`.

[76] https://x.com/OpenAIDevs/status/1735730662362189872 (last accessed on 16.01.24).

[77] As implemented by the `evaluate` package, see https://huggingface.co/docs/evaluate/index.

	TriviaQA		CoQA	
	Textual	Semantic	Textual	Semantic
Random	.11 ±.08	.00 ±.08	.08 ±.12	.00 ±.12
Clustering	.39 ±.28	.60 ±.14	.47 ±.25	.70 ±.17

Table 6.2: Results of evaluation of found clusters on TriviaQA and CoQA, including one standard deviation. Textual refers to similarity scores computed using ROUGE-L, and semantic scores based on cosine similarities of question embeddings of the same cluster. Here, we use random comparisons between questions in the dataset as a baseline.

sentence embedding model to capture the meaning of a sentence, the expect the semantic similarity to be high when questions are similar in topic, but might differ in their choice of words. Since performing this evaluation on the entire dataset is computationally expensive, we approximate the score by using 5 pairwise comparisons per cluster, with 200 comparisons for ROUGE-L and 1000 for cosine similarity in total, respectively. As a control for our method (*clustering*), we also compute values between unrelated questions that are not in the same cluster (*random*).

Results. We show the results of this analysis in Table 6.2. We observe noticeable differences between the random baseline and the similarity for the clustering scores, both on a textual and semantic level. While there is smaller difference on a textual level due to the relatively similar wording of questions, the semantic similarity based on the encoded questions is very notable. We provide deeper analyses of this part in Appendix B.9, showing that this method creates diverse ranges of calibration confidence targets. This suggests two things: On the one hand, our proposed methodology is able to identify fine-grained categories of questions. On the other hand, the diversity in calibration targets shown in Appendix B.9 indicates that we detect sets of questions on which the LLM's accuracy varies—and that this variety should be reflected. We test the ability of different methods to do exactly this next.

6.2.2 Calibrating White and Black-Box Models

Next, we test whether auxiliary models can reliably predict the target LLM's confidence. We describe our experimental conditions below.

Evaluation metrics. Aside from reporting the accuracy on the question-answering task, we also report several calibration metrics, including the expected calibration error (ECE;Naeini et al., 2015) using 10 bins. In order to address any distortion of results introduced by the binning procedure, we use smooth ECE (smECE; Błasiok and Nakkiran, 2023), which avoids the binning altogether by smoothing observations using a radial basis function kernel. We also consider Brier score (Brier, 1950), which can be interpreted as mean-squared error for probabilistic predictions. We further show how indicative the predicted confidence is for answering a question incorrectly by measuring the AUROC. The AUROC treats the problem as a binary error detection task based on the confidence scores, aggregating the results over all possible decision thresholds. In each case, we report the result alongside a bootstrap estimate of the standard error (Efron and Tibshirani, 1994) estimated from 100 samples and test for significance using the almost stochastic order test (del Barrio et al., 2018a; Dror et al., 2019; Ulmer et al., 2022c) with $\tau = 0.35$ and a confidence level of $\alpha = 0.1$.

Baselines. To contextualize the auxiliary calibrator results, we consider the following baselines: We consider the raw (length-normalized) sequence likelihoods (Seq. likelihood) as well as variant using Platt scaling (Platt et al., 1999): Using the raw likelihood $\hat{p} \in [0, 1]$ and the sigmoid function σ, we fit two additional scalars $a, b \in \mathbb{R}$ to minimize the mean squared error on the validation set to produce a calibrated likelihood $\hat{q} = \sigma(a\hat{p} + b)$ while keeping all other calibrator parameters fixed. We also compare it to the recent method of *verbalized* uncertainty (Lin et al., 2022a; Tian et al., 2023), where we ask the model to assess its confidence directly. We do this by asking for confidence in percent (Verbalized %) and using a seven-point scale from “very low” to “very high”, and which is mapped back to numeric confidence scores (Verbalized Qual.). Where applicable, we also distinguish between baselines with and without chain-of-thought prompting (CoT; Wei et al., 2022). For our approach, we distinguish between confidence targets obtained through the procedure in Section 6.1.2 (clustering) and simply predicting whether the given answer is correct or incorrect (binary).

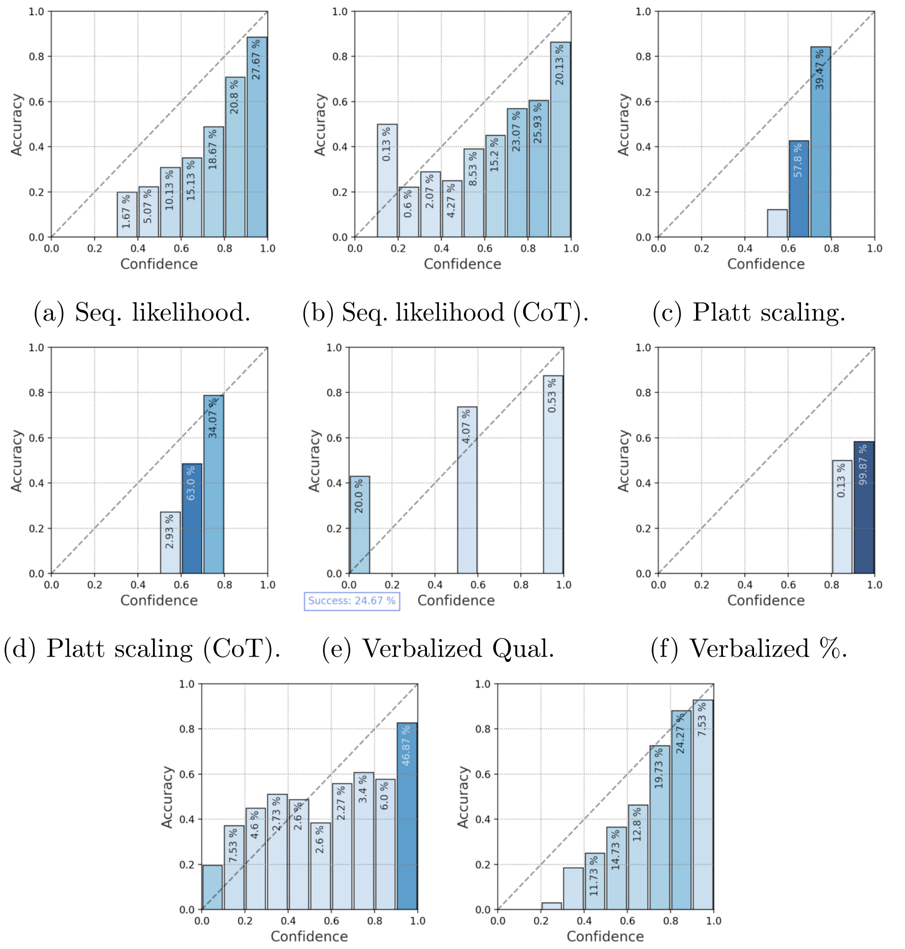
Results. Vicuna v1.5 7B achieves 58% accuracy on TriviaQA and 44% on CoQA, while GPT-3.5 obtains 85% and 55% accuracy,

		TriviaQA				CoQA			
	Method	Brier↓	ECE↓	smECE↓	AUROC↑	Brier↓	ECE↓	smECE↓	AUROC↑
Vicuna v1.5 (white-box)	Seq. like.	.22 ±.01	.05 ±.00	.03 ±.00	.79 ±.01	.32 ±.01	.08 ±.00	.08 ±.00	.69 ±.01
	Seq. like. (CoT)	.25 ±.01	.04 ±.00	.04 ±.00	.70 ±.01	.35 ±.01	.04 ±.00	.05 ±.00	.61 ±.01
	Platt	.24 ±.00	.08 ±.00	.07 ±.00	.70 ±.01	.30 ±.00	.03 ±.00	.03 ±.00	.69 ±.01
	Platt (CoT)	.24 ±.00	.12 ±.00	.11 ±.00	.79 ±.01	.30 ±.00	.02 ±.00	.02 ±.00	.61 ±.01
	Verb. Qual.	.38 ±.03	.02 ±.00	.02 ±.00	.62 ±.03	.45 ±.01	**.00** ±.00	**.00** ±.00	.48 ±.01
	Verb. Qual. (CoT)	.39 ±.02	**.01** ±.00	**.01** ±.00	.60 ±.02	.45 ±.01	**.00** ±.00	**.00** ±.00	.48 ±.01
	Verb. %	.39 ±.01	.38 ±.00	.27 ±.00	.52 ±.01	.49 ±.01	.48 ±.00	.32 ±.00	.53 ±.01
	Verb. % (CoT)	.39 ±.01	.38 ±.00	.26 ±.00	.49 ±.01	.48 ±.01	.06 ±.00	.06 ±.00	.55 ±.01
	Aux. (binary)	.20 ±.01	.16 ±.01	.15 ±.01	.81 ±.01	.20 ±.01	.16 ±.01	.15 ±.01	**.82** ±.01
	Aux. (clustering)	**.18** ±.00	.09 ±.01	.09 ±.01	**.83** ±.01	**.18** ±.00	.04 ±.01	.04 ±.01	**.82** ±.01
GPT-3.5 (black-box)	Seq. like.	.15 ±.01	.04 ±.00	.04 ±.00	.69 ±.02	.29 ±.01	.11 ±.00	.11 ±.00	.70 ±.01
	Seq. like. (CoT)	.14 ±.00	.05 ±.00	.05 ±.00	.60 ±.02	.25 ±.00	**.01** ±.00	**.02** ±.00	.52 ±.02
	Platt	.15 ±.00	.04 ±.00	.04 ±.00	.69 ±.02	.26 ±.01	.03 ±.00	.03 ±.00	.70 ±.01
	Platt (CoT)	.15 ±.00	.12 ±.00	.12 ±.00	.60 ±.02	.25 ±.00	.06 ±.00	.06 ±.00	.52 ±.02
	Verb. Qual.	.14 ±.01	.07 ±.00	.04 ±.00	.61 ±.02	.27 ±.00	.07 ±.00	.05 ±.00	.52 ±.01
	Verb. Qual. (CoT)	.15 ±.00	.04 ±.00	.03 ±.00	.63 ±.02	.30 ±.01	.08 ±.01	.04 ±.00	.50 ±.01
	Verb. %	.13 ±.01	.01 ±.00	**.01** ±.00	.63 ±.02	.34 ±.01	.25 ±.00	.22 ±.00	.54 ±.01
	Verb. % (CoT)	.13 ±.01	**.00** ±.00	**.01** ±.00	.63 ±.02	.37 ±.01	.09 ±.01	.06 ±.00	.49 ±.02
	Aux. (binary)	.14 ±.00	.14 ±.01	.14 ±.01	.65 ±.02	.19 ±.01	.13 ±.01	.13 ±.01	**.81** ±.01
	Aux. (clustering)	**.12** ±.01	.06 ±.01	.06 ±.01	**.72** ±.02	**.18** ±.00	.02 ±.01	**.02** ±.00	**.81** ±.01

Table 6.3: Calibration results for Vicuna v1.5 and GPT-3.5 on TriviaQA and CoQA. We bold the best results per dataset and model, and underline those that are statistically significant compared to all other results assessed via the ASO test. Results are reported along with a bootstrap estimate of the standard error.

respectively.[78] We present the calibration results in Table 6.3. APRICOT 🍑 achieves the highest AUROC in all settings and among the lowest Brier scores and calibration errors. On the latter metric, verbalized confidence beats our method, but often at the cost of a higher worst-case calibration error and lower AUROC. The effect of CoT prompting on calibration, however, remains inconsistent across different baselines. Lastly, APRICOT 🍑 with clustering beats the use of binary targets for Vicuna v1.5 and GPT-3.5 on both TriviaQA and CoQA. We also juxtapose reliability diagrams for the different methods for Vicuna v1.5 on TriviaQA in Figure 6.4 (we show the other reliability diagrams,

[78] We use the same heuristic based on thresholded ROUGE-L scores as in Section 3.2.3 or Kuhn et al. (2023) to determine whether an answer is correct. Since GPT-3.5 is a closed-source model, it is hard to say whether the higher accuracy scores are due to better model quality, test data leakage, or overlap in questions in the case of TriviaQA (Lewis et al., 2021).



(a) Seq. likelihood. (b) Seq. likelihood (CoT). (c) Platt scaling.

(d) Platt scaling (CoT). (e) Verbalized Qual. (f) Verbalized %.

(g) Auxiliary (binary). (h) Auxiliary (clustering).

Figure 6.4: Reliability diagrams for our different methods using 10 bins each for Vicuna v1.5 on TriviaQA. The color as well as the percentage number within each bar indicate the proportion of total points contained in each bin.

including for GPT-3.5, in Appendix B.10). Here it becomes clear that verbalized uncertainties approaches usually do not emit a wide variety of confidence scores. This is in line with observations by Zhou et al. (2023), who hypothesize the distribution of expressions generated by verbalized uncertainty heavily depend on the mention of e.g. percentage values in the model's training data. While Figure B.23 shows that GPT-3.5 provides more variety in this regard, the overall phenomenon persists.

Consistency of Verbalized Uncertainty. While verbalized uncertainties often perform well according to calibration error,

	Vicuna v1.5		GPT-3.5	
Method	TriviaQA	CoQA	TriviaQA	CoQA
Verb. Qual.	.19	.66	1.00	1.00
Verb. Qual. (CoT)	.25	.73	1.00	1.00
Verb. %	1.00	.99	1.00	1.00
Verb. % (CoT)	1.00	.99	.99	.58

Table 6.4: Consistency of verbalized uncertainty methods for Vicuna v1.5 and GPT-3.5 on TriviaQA and CoQA.

these results have to be taken with a grain of salt: Especially for the relatively small 7B Vicuna v1.5 model, the generations do not always contain the desired confidence expression, as visible by the low consistency in Table 6.4. CoT prompting seems to increase the success rate of verbalized uncertainty, and the additional results on GPT-3.5 suggests that this ability might also be dependent on model size. But even when taking the generated confidence expression, their ability to distinguish potentially correct from incorrect LLM responses remains at or close to random level. This suggests that due to the skewed distribution of confidence expressions, they can only be well-calibrated on datasets which are easy for the underlying model, which, naturally, is not known a priori. Next, we conduct some additional analyses based on the clustering-based variant of our method.

6.2.3 Ablation Study

The previous results pose the question of which parts of input the auxiliary model actually learns from. So, analogous to the different prompting strategies in Figure 6.3, we explore different input variants: First, we test a question-only setting, where the target LLM's answer is omitted completely. We also test the performance of the calibrator when given more information, for instance the model answer with and without chain-of-thought prompting, which could potentially expose flaws in the LLM's response.[79] Finally, we also expose the verbalized uncertainty of the LLM to the calibrator.

Results. We show these results in Table 6.5 in Appendix B.10. Interestingly, we can observe that even based on the question to

[79] Based on the recent study by Turpin et al. (2023), we assume that CoT does *not* expose the LLM's actual reasoning. Nevertheless, it provides more context about the given answer.

	Auxiliary Model Input				TriviaQA				CoQA			
	Quest.	Ans.	CoT	Verb.	Brier↓	ECE↓	smECE↓	AUROC↑	Brier↓	ECE↓	smECE↓	AUROC↑
Vicuna v1.5 (white-box)	✔	✗	✗	✗	.21 ±.00	**.07** ±.01	**.06** ±.01	.74 ±.01	.22 ±.00	**.03** ±.01	**.03** ±.00	.70 ±.01
	✔	✔	✗	✗	**.18** ±.00	.09 ±.01	.09 ±.01	**.83** ±.01	**.18** ±.00	.04 ±.01	.04 ±.01	**.82** ±.01
	✔	✔	✗	**Qual.**	**.18** ±.00	.08 ±.01	.08 ±.01	.82 ±.01	.19 ±.00	.04 ±.01	.04 ±.01	.79 ±.01
	✔	✔	✗	%	**.18** ±.00	**.07** ±.01	.07 ±.01	.82 ±.01	**.18** ±.00	**.03** ±.01	**.03** ±.01	.80 ±.01
	✔	✔	✔	✗	.19 ±.01	**.07** ±.01	.07 ±.01	.80 ±.01	.21 ±.00	.04 ±.01	**.03** ±.01	.74 ±.01
	✔	✔	✔	**Qual.**	.19 ±.00	.08 ±.01	.08 ±.01	.80 ±.01	.22 ±.00	**.03** ±.01	**.03** ±.01	.70 ±.01
	✔	✔	✔	%	**.18** ±.00	**.07** ±.01	.07 ±.01	.81 ±.01	.20 ±.00	**.03** ±.01	**.03** ±.00	.75 ±.01
GPT-3.5 (black-box)	✔	✗	✗	✗	**.12** ±.01	.05 ±.01	.05 ±.01	.71 ±.03	.21 ±.00	.03 ±.01	.03 ±.01	.72 ±.01
	✔	✔	✗	✗	**.12** ±.01	.06 ±.01	.06 ±.01	.72 ±.02	**.18** ±.01	.04 ±.02	.04 ±.02	**.82** ±.02
	✔	✔	✗	**Qual.**	**.12** ±.01	**.03** ±.01	**.03** ±.01	.72 ±.03	**.18** ±.01	**.02** ±.01	**.02** ±.00	.80 ±.01
	✔	✔	✗	%	**.12** ±.01	**.03** ±.01	**.03** ±.01	.72 ±.02	**.18** ±.00	.04 ±.01	.03 ±.00	.80 ±.01
	✔	✔	✔	✗	**.12** ±.01	.06 ±.01	.06 ±.01	.72 ±.02	.21 ±.00	.03 ±.01	.03 ±.01	.72 ±.01
	✔	✔	✔	**Qual.**	**.12** ±.01	.04 ±.01	.04 ±.01	**.73** ±.02	.21 ±.00	.04 ±.01	.04 ±.01	.72 ±.01
	✔	✔	✔	%	**.12** ±.01	.04 ±.01	.04 ±.01	.64 ±.02	.21 ±.00	**.02** ±.01	**.02** ±.00	.72 ±.01

Table 6.5: Calibration results for Vicuna v1.5 and GPT-3.5 on TriviaQA and CoQA using the auxiliary (clustering) method. We bold the best results per dataset, method and model.

the LLM alone, APRICOT 🍑 can already achieve respectable performance across all metrics. This suggests that the calibrator at least partially learns to infer the difficulty of the LLM answering a question from the type of question alone. Nevertheless, we also find that adding the LLM's actual answer further improves results, with additional gain when using CoT prompting. In some cases, the calibration error can be improved when using the LLM's verbalized uncertainties; in this sense, we can interpret the role of the calibrator as mapping the model's own assessment to a calibrated confidence score.

6.3 Discussion

Despite the difficulty of predicting the LLM's confidence from its generated text alone, our experiments have shown that APRICOT 🍑 can be used to produce reasonable scores even under these strict constraints. We showed in the past sections that the auxiliary model can be finetuned to learn from multiple signals. On the one hand, the auxiliary calibrator learns a mapping from a latent category of question to the expected difficulty for a target LLM. On the other hand, including the answer given through CoT prompting and including the LLM's own assessment of its uncertainty helped to further improve results. While sometimes beaten in terms of calibration error, our method consistently outperforms our baselines in error detection AUROC, meaning that it can provide the best signal to detect wrong LLM answers. Compared to other approaches, this yields some desirable properties: APRICOT 🍑 is available when sequence likelihood is not; it is more reliable than verbalized uncertainty; and it only needs a light finetuning once, adding negligible inference overhead. Compared to other methods such as Kuhn et al. (2023); Lin et al. (2023) in Section 2.3, it also does not require more generations for the same input, reducing the more expensive LLM inference costs.

Limitations. While yielding generally positive results in our case, the clustering methodology from Section 6.1.2 requires access to a sufficiently expressive sentence embedding model and a large enough number of data points. When this is not given, we show that the binary approach—tuning the auxiliary model to predict errors—is a viable alternative. As any neural model, the auxiliary calibrator is vulnerable to distributional shift and out-of-distribution data. Further research could help to understand how this issue can be reduced and which parts of the input the model identifies to predict confidence scores in order to unveil potential shortcut learning (Du et al., 2023). Our experiments focused on open-ended question-answering tasks, which provide a fairly easy way to check answer correctness. In other types of language generation such as summarization, translation or open text generation, this notion of correctness is not given.

6.4 Summary

In this chapter, we presented APRICOT 🍑, a general method to obtain confidence scores from any language model on the input and text output alone. We showed that it is possible to compute calibration targets through the clustering of question embeddings. Through the subsequent finetuning of a smaller language model, we then outperform other methods to distinguish incorrect from correct answers with competitive calibration scores, on different models and datasets. While we only presented a first, more fundamental version this approach in this work, it lends itself naturally to a whole body of research that aims to improve the calibration of pretrained language models (Desai and Durrett, 2020; Jiang et al., 2021; Chen et al., 2023b). Lastly, future studies might also investigate the uncertainty of the auxiliary model itself and use techniques such as conformal prediction in Section 2.2.1 to produce estimates of LLM confidence *intervals*.

7 Discussion

> “*When Sha Monk opened up a scroll of scripture that the other two disciples were clutching, his eyes perceived only snow-white paper without a trace of so much as half a letter on it. Hurriedly he presented it to Tripitaka, saying, ‘Master, this scroll is wordless!’ Pilgrim also opened a scroll and it, too, was wordless. Then Eight Rules opened still another scroll, and it was also wordless. ‘Open all of them!’ cried Tripitaka. Every scroll had only blank paper.*”
>
> —*The Journey to the West* (西游记), Ch. 94, as translated and edited by Anthony C. Yu (1977).

The last chapters have explored the various different definitions of and perspectives on uncertainty and how they materialize in the fields of machine learning and natural language processing. Despite the usefulness of uncertainty quantification for a whole spectrum of applications (Section 2.6) and its importance to avoid negative outcomes and to build trust in automation (Section 2.4), a somewhat fractured research landscape emerges: Uncertainty still remains a very under-defined and under-researched topic, especially in natural language processing. Uncertainty within the experimental pipeline often stays unaddressed or outright ignored; Uncertainty modeling poses a challenge under the current large language model paradigm and the successes and failures of uncertainty quantification are equally poorly understood. The efforts described in Chapters 3 to 6 can only work as a step to mitigate this fact, and thus dedicate this chapter to revisit the initial research goals defined in this thesis, and discuss a number of fundamental open questions and research directions.

7.1 Discussion of Research Questions

This thesis gave an overview over different notions of uncertainty from the perspectives of statistics, linguistics, deep learning and NLP in Chapter 2, also discussing how uncertainty can be communicated and how it interacts with human-AI trust. The influence of uncertainty on the experimental pipeline was analyzed in Chapter 3, where we could see how more careful experimental design allows to quantify uncertainty in results, reduce it, and even open up new avenues for modeling it. Some of the limits of uncertainty quantification for text classification were demonstrated in Chapter 4 using the theoretical case of ReLU networks and a large variety of different models applied to text classification tasks in English, Danish and Finnish. Lastly, non-exchangeable conformal prediction enables us to develop a method to obtain calibrated token sets for generation in Chapter 5 and APRICOT 🍑, a method to obtain calibrated confidence scores from black-box LLMs in Chapter 6. Based on this research, we now return to the research questions posed in Section 1.4 and discuss them in turn.

🔎 **RQ1**: How can uncertainty in NLP be characterized?

In Chapter 2 we discussed the multi-faceted views on uncertainty from a variety of perspectives, all of which coalesce in modern NLP applications. This includes the linguistic uncertainties present in the input data, interacting with the statistical uncertainties lingering in the modeling aspect.

Linguistically, uncertainty materializes as an inherent property of language in the form of underspecification, ambiguity and vagueness (Section 2.1.3), but also as a tool for humans to express their state of knowledge about the world (Section 2.1.4; this can also be used by language models to communicate uncertainty, see Section 2.5). Statistically, uncertainty is treated differently in the frequentist and Bayesian school of thought: Frequentists see probabilities as the relative frequency of an event under continued repetitions of an experiment. Bayesians interpret them as a degree of belief, with the parameter of interest turning from an unobserved constant into a random variable. Both perspectives are echoed in the corresponding neural approaches: Calibration techniques and conformal prediction on the one hand allow us to create confidence scores that reflect the correctness of the model, or prediction sets contain the ground truth in expectation. Approximating the neural weights posterior or parameterizing higher-order distributions on the other hand permit a decoupling of different notions of uncertainty.

The latter notions mostly refer to *predictive* uncertainty and are for example quantified in terms such as the total, data, model and distributional uncertainties. As Baan et al. (2023) point out, these can be seen as a spectrum, in contrast to a fixed set of discrete categories. This means that steps like data collection can be a source of model uncertainty when data is scarce, and can be reduced when more data is collected. However it can also produce data uncertainty which, in some instances, can be reduced through e.g. better annotation guidelines. In this light, the choice of method can be informed by the kind of uncertainty most useful to the problem at hand, and if necessary and possible, the experimental pipeline can be adapted to reduce uncertainty further or to enable better modeling of it (see next 🔎 RQ2). For active learning for instance, we might care most about epistemic or distributional uncertainty and therefore refer to Bayesian or evidential methods, while for error detection we might be satisfied with easy-to-implement estimators of total uncertainty.

It should be noted though that almost all methods discussed so far quantify uncertainty statistically rather than linguistically. While verbalized uncertainty (Section 2.5) is a step towards expressing uncertainty in words, it (thus far) ignores the rich shades of meaning that are at a human speaker's disposal (Figure 2.6). Communicating uncertainty to humans can be challenging (Section 2.4), so more natural verbalized uncertainty could prove to be a fruitful avenue of research.

🔎 **RQ2**: How can choices in experimental design help to reduce and quantify uncertainty?

In Chapter 3, we discussed the role of uncertainty in experimental design in NLP. There, we argued that careful data collection can help to reduce uncertainty caused by noise, and enable new modeling options through multiple annotations. Furthermore, hypothesis testing can help to quantify the uncertainty in results and aid model selection.

Uncertainty manifests in different stages of the experimental process and is often overlooked outside of the modeling stage; however, steps that are undertaken to increase reproducibility can help to rein in uncertainty and open modeling options. In NLP, this is exemplified by publishing all instance annotations (instead of an aggregate) and embracing human disagreements which arise from the ambiguities in language (Section 2.3; Plank, 2022; Baan et al., 2023). As we discuss in Section 7.2, this could for instance be combined with recent advances in evidential deep learning to learn higher-order distributions (Section 2.2.3).

Additionally, comparing different models, prompts or other settings can be difficult due to the non-linear nature of neural network and their increasing model sizes. In Section 3.2.1, we showed how to quantify this uncertainty in modeling results using the ASO test. As the test is non-parametric, we do not require any knowledge of the underlying distribution of scores. In the case study in Section 3.2.3, we furthermore demonstrated that even though modern LLMs tend to be pretrained, monolithic models, we can perform statistical hypothesis testing by obtaining observations from different prompts and thereby assessing their robustness (Mizrahi et al., 2024; Sclar et al., 2023). We also formalized the different distributions that are compared—in the LLM setting for instance, we keep the model architecture, pretraining data and hyperparameters constant while varying other factors such as prompt design and generation hyperparameters. Generally speaking, all of these settings vary a certain number of variables on which the output is conditioned on, while keeping others fixed. Although many variations of this setup are plausible, we believe it is important to make underlying assumptions more explicit and vary as many variables as feasible in order to arrive at a well-rounded estimate of model performance.

🔎 **RQ3**: How do inductive model biases influence uncertainty quantification?

Inductive biases describe the modeling assumptions present in a model's architecture and training procedure. As we saw in Chapter 4, this can have unintuitive effects on the efficacy of uncertainty estimates, where models may act confidently when faced with OOD inputs.

Many methods for uncertainty quantification equip a model with some sort of metric that operates on the model's output and translates it into a usually scalar measure of its uncertainty. While these have some expected or desired behaviors—such as the predictive entropy being high on OOD data—this is often not true in practice. This was illustrated for instance using ReLU networks in Section 4.1: Due to the inductive bias of the architecture, the network induces linear decision regions in the feature space, leading uncertainty metrics to provably converge to fix points in the limit (instead of being sensitive to the degree of familiarity with an input).

One might criticize the argument about ReLU networks for being too simplistic, since modern deep learning architecture are much more complex; and while it is true that this fact prevents similar proofs, we empirically identified similar problems on a large variety of text classification models in Section 4.2. We explicitly tested a low-resource setting (simulated for English), where training data is scarce and behavior on OOD might be unreliable. By testing on OOD test sets, we could show that similar failures occur in practice and that uncertainty measures are unable to effective distinguish in-distribution from foreign data.

How can we explain this behavior? One possible hypothesis is to look at this problem through the lens of the information bottleneck principle (Tishby et al., 2000; Tishby and Zaslavsky, 2015; Saxe et al., 2018): Neural predictors often map input representations into lower-dimensional latent spaces. This way, they are incentivized during training firstly to recover the correct prediction, and secondly to compress the input in a way that supports the first goal. Intuitively, we can assume that this learned compression will favor features that are most useful to the predictive task, not necessarily ones that are useful to indicate uncertainty. Indeed, some works in anomaly detection have noted that neural models might fail to encode novel, unseen features that might indicate that a test point is out-of-distribution (Dietterich and Guyer, 2022; Sivaprasad and Fritz, 2023). In addition, other works have noted how in- and out-of-distribution features overlap in latent space (van Amersfoort et al., 2021). But these features are exactly what should indicate model uncertainty, since the model is likely to be misspecified on points different from the training distribution! This means that this dynamic might make uncertainty quantification unreliable in cases where we cannot obtain good estimates of epistemic uncertainty, or where epistemic uncertainty accounts for a large portion of the total uncertainty. In the theoretical analysis in Section 4.1, uncertainty estimates can still be useful in regions of class overlap (hence, aleatoric uncertainty), but fail to be informative in regions without model training data due to their convergence to fix points. In the empirical study in Section 4.2, we observe that the quality of uncertainty estimates can decrease as we add more training data, potentially due to the selective compression phenomenon. From this we can deduce that the inductive biases of standard architectures are insufficient for reliable uncertainty quantification, and better inductive biases are needed.

One possible solution of this lies in directly modeling the data density. Language models do this already by assigning probabilities to entire sequences; however, Section 6.2.2 and Kumar and Sarawagi (2019) showed that sequence likelihoods are insufficient for error prediction, and other studies such as Ren et al. (2022) have demonstrated their failure on OOD detection. This can be explained by the fact that language models are trained on only a single sequence in a combinatorically large space of possible continuations. This automatically implies a sort of data scarcity, where the model fails to adequately capture the paraphrasticity of language (see Section 2.1.3). LeBrun et al. (2022) discovered how language models tend to overestimate the probability of frequent sequences and underestimate the ones coming from the tail end of the sequence distribution, with similar findings by Ilia and Aziz (2024); Liu et al. (2024a).[80]

As another approach to better inductive biases for UQ, one might choose to model the distribution of latent representations instead. This is for instance done through normalizing flows in the case of posterior networks (Section 2.2.3) or some methods regarding direct uncertainty prediction (Section 2.2.4). But since these components are trained on the latent encodings of an underlying model, they can only learn the distribution of latent features that are learned by the main model, and might thus fall into the same trap of not modeling features indicative of model uncertainty that were "compressed away". This can explain why the DDU Bert in Section 4.2.5 does not attain its best results on OOD detection through the log probability of its latent density estimator, and why posterior networks have been shown to not always detect OOD reliably (Kopetzki et al., 2021).

🔎 **RQ4**: How can we address some of the challenges of uncertainty quantification in NLP?

In this thesis, we addressed multiple of the challenges that we laid out in Section 1.3, including data scarcity and sequentiality. For clarity, we will discuss them here in turn and the corresponding insights gained from this work.

Challenges of Natural Language. In this thesis, we mainly worked towards solving two of the challenges that come with natural

[80] This phenomenon might also be the culprit behind the inadequacy of sampling from the mode in NLG, see for instance Eikema and Aziz (2020); Holtzman et al. (2020); Eikema (2024).

language data, namely its diversity and sequentiality. On the one hand, Section 4.2 tested different uncertainty methods for text classification on three different languages and OOD test sets that introduce novel domains. While general trends are visible across all settings, we can also see that the best uncertainty quality in terms of model and corresponding metric differs across datasets. This suggests that there might be complex underlying interactions between the model and the types of uncertainty that OOD data evokes in it, the uncertainty quantification method, and language-specific characteristics.[81] For the non-exchangeable conformal language generation in Chapter 5, we also tested on German and Japanese as different source language for the machine translation task. We measured coverage, namely whether conformal prediction sets contain the ground truth continuation, and translation quality, but found only minor differences between languages, with similar trends across tested methods. Importantly, this method addresses the sequentiality issue in natural language: Even though it is possible to conformalize language generation on a sequence-level where the i.i.d. assumption is maintained (see Quach et al., 2023), we were able to provide a method on a token-level that provides a well-motivated framework. This is different compared to cases like Ravfogel et al. (2023), who operate on a token-level but have to make strong assumptions about the underlying data that might not be realistic in practice.

Data Scarcity. In Section 4.2, we explicitly tested low-resource settings by using under-resourced languages such as Finnish and Danish, and by testing the relationship between training set size and uncertainty quality. Unsurprisingly, we showed that task performance increases with the amount of data. More surprisingly, we showed that increased amount of training data can have adverse effects on uncertainty quality on OOD inputs, for possible reasons we discussed in the answer for 🔎 RQ3.

Trust & Safety. Firstly, this thesis introduced non-exchangeable conformal language generation in Chapter 5, which provides a way to produce sets of token for generation with conformal guarantees. Similarly to standard prediction sets in Section 2.2.1, other ways of truncating the predictive distribution over tokens do not provide any guarantees of containing the correct continuation. Nev-

[81] The ability to model linguistic idiosyncrasy's can to some degree also be influenced by the quality of tokenization and therefore the models' uncertainty. For investigation into the first point, refer e.g. to Graën et al. (2018); Virtanen et al. (2019); Singh et al. (2019); Rust et al. (2021); Pfeiffer et al. (2021); Mielke et al. (2021); Maronikolakis et al. (2021).

ertheless, these prediction sets can be conformalized through our calibration method that utilizes information from nearest neighbors from a datastore. Not only does the generation process now (approximately) fulfill conformal guarantees, this also opens up new possibilities through the extension of (non-exchangeable) conformal risk control (Angelopoulos et al., 2023; Farinhas et al., 2024): Future approaches could provide bounds on a wider family of functions, more instance measuring toxicity, veracity or alignment with human values, similar to the works of Mohri and Hashimoto (2024); Gui et al. (2024). The latter has already been explored as an on-the-fly procedure (albeit, not conformal) instead of an additional finetuning stage (Yang and Klein, 2021; Qin et al., 2022; Mudgal et al., 2023; Gao et al., 2024a). The fact that conformal methods can provide statistical guarantees for otherwise unwieldy language models has also spurred additional work on the subject, for instance conformalizing generation on a sequence-level (Quach et al., 2023), for prompt selection (Zollo et al., 2023), conditional computation (Schuster et al., 2022; Ren et al., 2023), planning for LLM agents (Liang et al., 2024), and for black-box models (Su et al., 2024). Secondly, for the most restrictive setup in which we are dealing with a black-box LLM and only have access to its input and generated text, we proposed APRICOT 🍑 in Chapter 6. We demonstrated that even in this context, using a secondary auxiliary model enables us to predict the target LLMs confidence reliably. We also showed that by clustering the latent presentation of inputs, we can use these clusters to obtain more fine-grained information about the expected performance of the LLM on a certain category of inputs. While we leave further exploration of this question to future work, it is intuitive to assume that this very extreme setup has limits on the reliability of confidence estimates. In this way, we can order different methods on a spectrum from full access to the model, including latent representations, to access to logits and the predictive distribution to text-only access. Some works have found that OOD inputs are detectable based on the model's hidden representations (Yoo et al., 2022; Ren et al., 2022), with similar insights for hallucination detection (Ferrando et al., 2022; Guerreiro et al., 2023a; CH-Wang et al., 2023; Duan et al., 2024) and general uncertainty quantification (Vazhentsev et al., 2023; Liu et al., 2024b), potentially suggesting a link back to the discussion about encoded and undecoded latent features from the previous 🔎 RQ3.

7.2 Open Questions & Future Research Directions

The answers to 🔎 RQs1 to 4 can only provide partial steps towards solving any of these complex questions. As this thesis has argued, the topic of uncertainty quantification in NLP lies in the intersection of multiple different fields such as statistics, linguistics and deep learning. It has only recently started to garner more attention, as for instance demonstrated by the first UncertaiNLP workshop (Vázquez et al., 2024), related surveys (Baan et al., 2023; Hu et al., 2023b; Geng et al., 2023; Campos et al., 2024) or other dissertations (He, 2024). This creates ample space for future research, which we outline next.

7.2.1 Modeling Uncertainty

One focus of research about uncertainty in deep learning is—and has been—its modeling. Despite the manifold of works in this direction however, a number of many open directions of research remain. This includes everything from the modeling uncertainty on different input scales, obtaining guarantees, and how to properly represent and explain it.

Influence of Experimental Design. Chapter 3 has argued how careful experimental design can reduce or help to quantify uncertainty, for instance by providing clearer annotation guidelines or model selection through statistical hypothesis testing. An often overlooked aspect is how retaining multiple human labels per training instance also opens up new avenues for better modeling of uncertainty and paraphrasticity (Plank, 2022; Baan et al., 2022, 2023).

Uncertainty with Guarantees. Pivotally, uncertainty quantification can only increase trust in ML systems when the estimate of uncertainty is itself reliable. As for instance Dhuliawala et al. (2023) showed, unreliable estimates can lead to a loss of trust in the model that can be hard to recover from. Thus, conformal prediction currently is a very promising research direction, since it supplies statistical guarantees about predictions that are furthermore agnostic to the underlying predictor. This flexibility has enables the flurry of conformal works in NLP (e.g. Schuster et al., 2022; Ravfogel et al., 2023; Quach et al., 2023; Zollo et al., 2023; Su et al., 2024; Ulmer et al., 2024c; Campos et al., 2024). Conformal prediction however comes with two caveats: Coverage is only guaranteed in expectation, and is *marginal* rather than *conditional*, i.e. the guarantee is $p(y' \in \mathcal{C}(\mathbf{x}')) \geq 1 - \alpha$ rather than $p(y' \in \mathcal{C}(\mathbf{x}') \mid \mathbf{x}') \geq 1 - \alpha$. Unfortunately, conditional coverage is generally deemed unachievable under finite samples, with the guarantee approximately being fulfilled in some situations (Vovk, 2012; Foygel Barber et al., 2021; Gibbs et al., 2023). Other ways to circumvent this issue lie in partitioning the dataset (similar to the binning in the ECE, see Feldman et al., 2021; Gibbs et al., 2023; Jin and Ren, 2024) or conditioning on the label y^* instead of the input (see mondrian conformal predictors; Vovk et al., 2005). Therefore, future research could investigate conformalizing other uncertainty methods or extending existing guarantees.

Hierarchical Uncertainty. Compared to other input modalities such as images, uncertainty in NLP exists on different scales. Starting from (subword-)token uncertainty, uncertainty can also exist on a sequence, utterance, or paragraph or even dialogue-level. So far, most uncertainty quantification techniques operate on a token-level or sequence-level, with pioneering work on higher scales such as the dialogue-level (Sicilia et al., 2024). While there are some theoretical frameworks like Malinin and Gales (2021) to model how uncertainty from tokens affects the uncertainty in sequences, this is only given for certain metrics. Therefore, an open question remains how to estimate uncertainty on these different levels and how uncertainty can be decomposed into smaller units.

Representing Uncertainty. In this thesis, we have mostly focused on representing uncertainty in the form of single scalars or prediction sets. However, uncertainty can also be represented in many other ways, for instance in the form of a posterior distribution or the highest density interval in Section 2.1.2, uncertainty in the latent space (Kingma and Welling, 2014; Rezende et al., 2014; Daxberger and Hernández-Lobato, 2019; Kong et al., 2020b; Miani et al., 2022), or even linguistically (see discussion in Section 7.2.4). The representation of uncertainty should therefore not be overly restrictive, embrace the richness in options and explore new representations.

Quantifying Human Uncertainty. Most of this thesis was focused on modeling and quantifying the uncertainty in models operating on language data, but one might also want to model the human uncertainty underlying the data directly. First advances in this direction have been made by estimating the uncertainty in human labels (Northcutt et al., 2021; Jiang et al., 2023b; Gruber et al., 2024), analyzing annotator disagreement (Baan et al., 2022,

2024) or comparing the variability of humans to that of NLG systems (Giulianelli et al., 2023; Lee et al., 2023; Ilia and Aziz, 2024). Furthermore, a number works try to model the uncertainty in humans using neural language models (Hu et al., 2023a) or try to detect linguistic uncertainty in text (Szarvas et al., 2012; Vincze, 2014; Kolagar and Zarcone, 2024).

Explaining Uncertainty. The answer to 🔎 RQ3 suggest a hypothesis with which the general behavior of uncertainty is influenced by neural inductive biases. Nevertheless, there also lies tremendous value in understanding how uncertainty actually arises for a specific input. This can for instance highlight erroneous or noisy parts of an input or help to understand model failure cases (see e.g. Xu et al., 2020 for an application to text summarization). To this extent, some works have began to apply interpretability techniques to understand predictive uncertainty, including Shapley values (Chen and Ji, 2022; Watson et al., 2024) or feature attribution methods (Bley et al., 2024).

7.2.2 Limits of Uncertainty Quantification

Another often overlooked aspect of uncertainty is defining or exploring the boundaries in which the model's uncertainty is expected to operate; this includes in particular cases in which uncertainty estimates themselves might be uncertain, ill-defined, limited, or reductive, and which are open for further exploration.

Limits of the Aleatoric–Epistemic Dichotomy. Uncertainty, in a statistical sense, is traditionally delineated along data (aleatoric) and model (epistemic) uncertainty (Hora, 1996; Der Kiureghian and Ditlevsen, 2009; Hüllermeier and Waegeman, 2021). However, recent works such as Baan et al. (2023); Gruber et al. (2023) have advocated to reject this dichotomy in favor of placing uncertainties and their sources on a spectrum. This dichotomy becomes blurred further when considering that more far-reaching decompositions are possible (for instance adding distributional uncertainty like in Section 2.2.3), and that estimates of epistemic uncertainty might be in themselves uncertain (Wimmer et al., 2023).

Higher Order Uncertainties. Evidential deep learning (Section 2.2.3) and credal learning (Section 2.2.4) offer methods to model higher-order probability distributions or sets and quantify their uncertainty. Having said that, evidential deep learning in particular has been criticized for not providing loss functions that can provably achieve well-behaved epistemic uncertainties in the model (Bengs et al., 2023), however alternative methods have been proposed for credal predictors (Hüllermeier et al., 2022; Sale et al., 2023a, 2024; Hofman et al., 2024).

Features for Uncertainty Quantification. The previously mentioned methods quantify uncertainty based on properties of the underlying probability distribution parameterized by a neural network. However, the considerations in 🔎 RQ3 might prompt one to consider whether this should be the only source from which we should deduce uncertainty. In the previous section we discussed for instance modeling uncertainty in the latent space, and Section 6.2.3 illustrated how, to some extent, we can infer uncertainty solely from the input to a model and train a secondary predictor to output uncertainty in a supervised learning task. Thus there remain many avenues to explore to find the best features that can be used to obtain uncertainty estimates, which are already being explored by works such as Fathullah et al. (2024); Liu et al. (2024b).

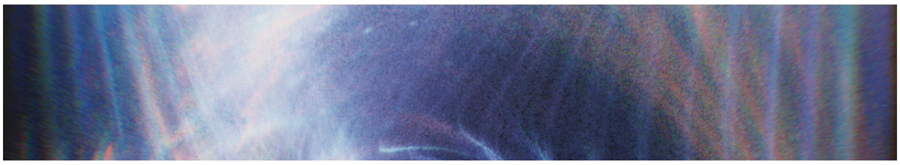
7.2.3 Evaluating Uncertainty

One common conundrum in the research surrounding uncertainty quantification is the lack of ground truth about a predictors uncertainty. Therefore—and in this regard Chapters 4 and 6 are no different—one has to instead defer to approximations and proxy tasks. For frequentists methods like confidence scores we can measure calibration errors, but have to make do with binning, kernel estimators or other approximations. Otherwise we fall back other problems like error or OOD detection or measure correlations between predictive error and uncertainty. These analyses need to be multi-dimensional to be cogent and can be gamed; for example the SNGP Bert in Section 4.2.5 achieves high correlation between sequence uncertainties and loss by not converging properly, and verbalized uncertainty by GPT-3.5 in Section 6.2.2 is well-calibrated on TriviaQA since the dataset is too easy, despite only articulating the same (high) confidence scores.

Yet when multiple annotations are available, we can actually use this to our advantage to create a ground truth for uncertainty, as done for instance by Baan et al. (2022); Ilia and Aziz (2024). Here, the paraphrasticity of language can help to create ground truth distributions whose uncertainty can be measure and compared against.

7.2.4 Communicating Uncertainty

Communicating uncertainty is difficult—Section 2.5 described how communicating uncertainty to different social groups while being both understandable and precise is challenging, and how the process can affect human-machine cooperations in sometimes unintuitive ways. In this light, verbalized uncertainty (Section 2.3) seems like an attractive tool for humanly intuitive ways of expressing uncertainty. But the experiments in Section 6.2.2 and studies such as (Tian et al., 2023) exemplified that such expressions tend to display lopsided distributions of confidence that are not desirable. Zhou et al. (2023) show how this behavior might be rooted in the unequal distribution of these confidence expression (in their case, percentage values) in the training data. This is not to say that this approach is moribund: Works like Mielke et al. (2022); Stengel-Eskin et al. (2024) train language models to produce more complex verbalized expressions of uncertainty, and Section 2.1.4 outlines the richness of human uncertainty expressions that can serve as a guide for future research.



8 Conclusion

> “*These intelligent agents are the only way to sift through the oceans of data we are producing at an exponential rate [...]. It is important if you find this terrifying or wonderful because public sentiment drives education, investment and regulation. If people find the rapid advance of intelligent machines terrifying instead of wonderful it won't stop it, but it could make the outcome worse for us all.*”
>
> —Garry Kasparov in *Deep Thinking* (Kasparov, 2017).

On May 6th 2023, a document submitted to the United States District Court of the Southern District of New York (The United States District Court for the S.D.N.Y., 2013) reads:

> “*The Court is presented with an unprecedented circumstance. A submission filed by plaintiff's counsel in opposition to a motion to dismiss is replete with citations to non-existent cases. When the circumstance was called to the Court's attention by opposing counsel, the Court issued Orders requiring plaintiffs counsel to provide an affidavit [...]. Six of the submitted cases appear to be bogus judicial decisions with bogus quotes and bogus internal citations.*”

The document was submitted by the judge in the case of Roberto Mata versus the Columbian airline Avianca. As it was revealed later, the plaintiff's lawyers used OpenAI's ChatGPT to find other relevant cases for their argument, which turned out to be non-existent.[82] This curious case represents three different aspects about AI in modern society at once: Firstly, AI in general and LLMs specifically are increasingly permeating society and culture. This can be shown through their growing adoption (Humlum and Vestergaard, 2024), their impact on art (Zulić, 2019; Du, Wenda, and Han, Qing, 2021; Sivertsen et al., 2024) and by becoming an progressively political issue (Hovy and Spruit, 2016;

[82] See for example the corresponding articles by the Verge (https://www.theverge.com/2023/5/27/23739913/chatgpt-ai-lawsuit-avianca-airlines-chatbot-research) or the New York Times (https://www.nytimes.com/2023/05/27/nyregion/avianca-airline-lawsuit-chatgpt.html). Both were accessed last on 17-05-2024.

Mohamed et al., 2020; Zuboff, 2023; Devenot, 2023). Secondly, current language models are prone to producing hallucinations, i.e. seemingly plausible but fabricated generations. While detection and mitigation of hallucinations are very active areas of research (Ji et al., 2023b), some have argued that it is an unavoidable feature of current models (Kalai and Vempala, 2024; Xu et al., 2024). Thirdly, the way language models work remains too technical and opaque to most people and LLM-based chatbots are conceptualized as search engines rather than extremely powerful word predictors. This becomes even more blatant when examining the details of the above case through one of the lawyers' affidavit: In order to verify the veracity of the (later to be found fictitious) cited case studies, they asked ChatGPT questions such as "*Is varghese a real case*", to which the language model answered affirmatively.

The *bitter lesson* (Sutton, 2019) states that "general methods that leverage computation are ultimately the most effective, and by a large margin". In the past, it has proven time and time again that sophistication in AI research is outperformed by sheer scale. Which, given the content of thesis, prompts the question of whether research on UQ is necessary or yet another piece of unnecessary ornamentation on the road to more intelligent systems.

Do We actually Need UQ? Let us assume the role of a devil's advocate for a moment. In this position, we can pose several counter-arguments to the necessity of UQ, starting with

> "*Current cutting-edge models work so well that UQ is not necessary.*"

While it is true that the bitter lesson keeps materializing in current models, even an ever-increasing coverage of topics and tasks through larger amounts of training data does not shield them from an infinitely-large space of possible inputs, on which their behavior is hard to predict. This phenomenon is referred to as *model underspecification*. Furthermore, increasing generalization by obtaining more and more training data is expensive; estimations by works such as Villalobos et al. (2022) suggest that we are already starting to deplete the stock of high-quality language data to train on. Counter-strategies to this problem have been to simply allocate resources to human data creation,[83] to repeatedly use the

[83] See for instance reporting about OpenAI's strategy to employ workers in Kenya to create new training data and improve existing data quality, e.g. https://time.com/6247678/openai-chatgpt-kenya-workers/ (last accessed 19.05.2024).

same training data (Xue et al., 2024b) or to use synthetic training data, where the latter has shown mixed results (Guo et al., 2023; Alemohammad et al., 2023; Briesch et al., 2023; Bohacek and Farid, 2023; Gulcehre et al., 2023; Shumailov et al., 2023; Feng et al.; Ulmer et al., 2024b). However, this also ignores the inequality of available data in different languages (Singh et al., 2024b). Being able to guarantee robust model behavior on different topics, tasks and language this way thus appears unlikely.

"*Model capabilities have consistently improved with model size and the amount of available training data, and in the same way a model's uncertainty estimates will become more reliable by itself.*"

While there is some evidence that e.g. a model's calibration increases with the available training data (Dan and Roth, 2021; Chen et al., 2023b; Tian et al., 2023; Zhu et al., 2023; Ulmer et al., 2024a), one can hypothesize that the increased coverage of training cases simply enables the model to better learn the actual distributions over targets (be it class labels or token distributions) for the most frequent types of input. For LLMs, there is some evidence that verbalized uncertainty in its current form improves with model and training data size, but the distribution of uncertainty expressions still remains skewed (Tian et al., 2023; Ulmer et al., 2024a).

"*Smarter models will become better at admitting when they do not know an answer.*"

Compared to the previous question, here we wouldn't rely on additional uncertainty estimates to refuse a potential unreliable prediction, but assume that a smarter model would learn to refuse directly. We can reason through this argument by realizing that in order to achieve these model refusals, they would either have to be explicit contents of their training data, or be the result of of some subsequent finetuning / alignment process. The first case is unrealistic or at least conceptually misguided: We would like models to respond to certain instructions by admitting their ignorance because the answer would otherwise likely be incorrect, not because they learned a mapping from certain instructions to these admissions—in the end, we still want models to learn to solve a given task! This entails that such a behavior would be acquired during additional finetuning steps (instead of the pre-training phase, such as instruction finetuning, alignment using human feedback, etc.), but in order to do so, one requires knowledge about when these statements are necessary. This could come from signals from the model itself—however we have seen that models *do not always know when they do not know*—or from human or automatic evaluation, which seems infeasible to perform on a comprehensive scale. Thus, we can likely only adopt these behaviors for more common instructions, even though they would matter most on unseen or rare ones.

"*Current UQ quantification approaches are useless since they are not reliable themselves.*"

This is not an entirely unfair criticism, and we dedicated parts of Chapter 7 to the limits and failure cases of current UQ methods. One could explain the recent soaring in interest in conformal prediction methods that they, in contrast to their alternatives, can provide formal guarantees. Even though these might still be insufficient for many practical applications, we can expect future research to improve them further. Furthermore, there is a case to made where the overall utility of UQ with even somewhat deficient guarantees exceeds the loss in utility without any UQ whatsoever. Given this thought, one might wonder why we haven't seen wide-spread adoption of UQ techniques in user-facing applications.

What Hinders UQ in User-Facing Products? This point can only be answered speculatively, but what is true is that none of the large commercially available LLMs at the time of this writing offer any degree of uncertainty quantification.[84] One potential reason could be that there is simply no or not enough demand; this could be because models usually work sufficiently well for users on their specific use cases or that customers are not aware of the problem (or of UQ as a possible solution). Another reason could be that UQ in its current form does not work reliably enough and would expose a company to too many risks; an unreliable prediction that is accompanied with a high confidence value could potentially create PR and legal liability issues when found to have caused real-world harm.

How does UQ Relate to Current Developments in the Field? At the time of writing of the author's master thesis in 2019, the field of NLP was experiencing an acceleration. After the invention of the transformer two years prior (Vaswani et al., 2017), models like Bert (Devlin et al., 2019) and GPT-2 (Radford et al., 2019) were heralding a paradigm shift in the field, as increasingly large models were demonstrating hitherto unseen abilities. In this context, part of the conclusion of Ulmer (2019) reads

[84] This includes Anthropic, Cohere, OpenAI, Google and Mistral. OpenAI's API does allow access to token probabilities (https://x.com/OpenAIDevs/status/1735730662362189872, last accessed on 16.01.24), however they are not framed as confidence scores directly, confidence estimation is just mentioned as one possible application.

> *"On the flip side, these [language] models require huge amounts of data and computational resources. [...] This has several, worrying implications: First, with these resource requirements, scientific papers become hard to reproduce. These costs only allow training of these models in the context of well-funded institutions, namely top-tier universities and affluent tech giants. Secondly, the reliance on large-scale hardware produces a high electricity consumption along with a worrisome carbon footprint, which bears a certain irony: These models try to (loosely) imitate the human brain, a biological computer that is actually very energy efficient (Schwartz et al., 2019). Lastly, scaling up data sets and the number of parameters does not necessarily increase the semblance to human cognition."*

It is interesting to re-examine these thoughts in the light of current trends. First of all, the size of language models and their training set sizes has risen tremendously. Devlin et al.'s largest Bert model comprised 340 million parameters, and was trained on around 3.3 billion words. For comparison, the largest Llama 3 model comprises 90 billion parameters and was trained on 15 trillion tokens (AI@Meta, 2024), with GPT-4 rumored to be 1.76 trillion parameters large (The Decoder, 2023). The fact that GPT-4's parameter count is not public and that details about the training data for both GPT-4 and Llama 3 are unknown accentuate the most recent trend in language model development and echo some of the thoughts above: With a few exceptions such like OLMo (Groeneveld et al., 2024), it has become infeasible for non-industry actors to train language models from scratch. At the same time, companies have started to hide training details that they deem strategically important, hindering replication and research even when the final models become openly available. This also makes it hard to assess for which kind of inputs we can expect models to behave reliably. This is exacerbated by the fact that any semblance of human intelligence is still controversial—while recent models have displayed impressive abilities (Bubeck et al., 2023), some argue that outputs are "haphazardly stitch[ed] together sequences of linguistic forms [the language model] has observed in its vast training data, according to probabilistic information about how they combine, but without any reference to meaning" (Bender et al., 2021). The consequence of this is that language models might fail in ways that are unpredictable and unintuitive to humans. And as the introductory examples in this chapter and Chapter 6 show, the more convincing generations appear, the harder any failures become to spot.

Policy and Societal Implications. The increased adoption of AI models has prompted a response from different regulatory bodies. One instance of this is the EU AI act (Madiega, 2021). The act sorts different applications into a four tier system, ranging from minimal risk to unacceptable risk. While unacceptable risk applications are outright prohibited (e.g. social scoring systems, facial recognition etc.), there also exists a tier of high-risks systems with applications in law enforcement, education or medicine that are allowed under strict regulations. One prerequisite for high-risk systems is *human oversight*, meaning that the system can be "effectively overseen by natural persons during the period in which the AI system is in use" and to "prevent or minimize the risks to health, safety or fundamental rights that may emerge" (Article 14). It should be clear that techniques like anomaly detection and UQ can help to fulfill these criteria by deferring decisions to human overseers and flagging inputs on which the system could behave abnormally. Thus, in order to create commercial high-risk AI applications in the EU, the development of UQ methods with stronger guarantees might be one potential avenue. Similar policies are still pending in the United States, where the Biden administration enacted an executive order on the development and use of AI (Biden, 2023). In its opening paragraph, it states

> "*Harnessing AI for good and realizing its myriad benefits requires mitigating its substantial risks. This endeavor demands a society-wide effort that includes government, the private sector, academia, and civil society.*"

AI is a powerful technology, unfolding in an unequal world and already reshaping societies. As researchers, we can help advance directions like UQ alongside others such as generalization, bias mitigation, fairness, interpretability and many more in order to help mitigate the risk of modern AI systems, so that any transformation may be a positive one.

Bibliography

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. 2021. Deep Reinforcement Learning at the Edge of the Statistical Precipice. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 29304–29320. (Cited on page 80)

Eneko Agirre and Philip Edmonds. 2007. *Word Sense Disambiguation: Algorithms and Applications*, volume 33. (Cited on page 26)

Kabir Ahuja, Sunayana Sitaram, Sandipan Dandapat, and Monojit Choudhury. 2022. On the Calibration of Massively Multilingual Language Models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4310–4323. (Cited on page 59)

Lukas Aichberger, Kajetan Schweighofer, Mykyta Ielanskyi, and Sepp Hochreiter. 2024. How many Opinions Does Your LLM Have? Improving Uncertainty Estimation in NLG. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*. (Cited on page 61)

AI@Meta. 2024. Llama 3 Model Card. (Cited on pages 78, 138, and 174)

Laura Aina and Tal Linzen. 2021. The Language Model Understood the Prompt Was Ambiguous: Probing Syntactic Uncertainty through Generation. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 42–57. (Cited on page 60)

Adrian Akmajian, Ann K. Farmer, Lee Bickmore, Richard A. Demers, and Robert M. Harnish. 2017. *Linguistics: An Introduction to Language and Communication*. (Cited on page 26)

Ahmed M. Alaa and Mihaela van der Schaar. 2020. Discriminative Jackknife: Quantifying Uncertainty in Deep Learning via

Higher-order Influence Functions. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 165–174. (Cited on page 115)

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. 2023. Self-consuming Generative Models Go MAD. In *The Twelfth International Conference on Learning Representations*. (Cited on page 172)

Junaid Ali, Preethi Lahoti, and Krishna P. Gummadi. 2021. Accounting for Model Uncertainty in Algorithmic Discrimination. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 336–345. (Cited on page 67)

Hussam Alkaissi and Samy I. McFarlane. 2023. Artificial Hallucinations in ChatGPT: Implications in Scientific Writing. *Cureus*, 15(2). (Cited on page 123)

James Urquhart Allingham, Florian Wenzel, Zelda E. Mariet, Basil Mustafa, Joan Puigcerver, Neil Houlsby, Ghassen Jerfel, Vincent Fortuin, Balaji Lakshminarayanan, Jasper Snoek, Dustin Tran, Carlos Riquelme Ruiz, and Rodolphe Jenatton. 2022. Sparse MoEs meet Efficient Ensembles. *Transaction on Machine Learning Research*, 2022. (Cited on page 46)

P.C. Álvarez-Esteban, Eustasio del Barrio, Juan Antonio Cuesta-Albertos, and C. Matrán. 2017. Models for the Assessment of Treatment Improvement: The Ideal and the Feasible. *Statistical Science*, 32(3):469–485. (Cited on pages 85 and 86)

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete Problems in AI Safety. *ArXiv preprint*, abs/1606.06565. (Cited on page 3)

Jakob Smedegaard Andersen and Walid Maalej. 2022. Efficient, Uncertainty-based Moderation of Neural Networks Text Classifiers. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1536–1546. (Cited on page 69)

Jakob Smedegaard Andersen and Olaf Zukunft. 2022. More Sustainable Text Classification via Uncertainty Sampling and a Human-in-the-loop. In *International Conference on Agents and Artificial Intelligence*, pages 201–225. Springer. (Cited on page 69)

Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I. Jordan. 2003. An Introduction to MCMC for Machine Learning. *Machine learning*, 50:5–43. (Cited on page 42)

Anastasios Nikolas Angelopoulos and Stephen Bates. 2021. A Gentle Introduction to Conformal Prediction and Distribution-free Uncertainty Quantification. *ArXiv preprint*, abs/2107.07511. (Cited on pages 39, 124, and 128)

Anastasios Nikolas Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. 2023. Conformal Risk Control. In *The Twelfth International Conference on Learning Representations*. (Cited on pages 40, 138, and 164)

Anastasios Nikolas Angelopoulos, Stephen Bates, Michael I. Jordan, and Jitendra Malik. 2021. Uncertainty Sets for Image Classifiers Using Conformal Prediction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. (Cited on pages 128, 131, and 138)

Rushil Anirudh and Jayaraman J. Thiagarajan. 2021. Delta-UQ: Accurate Uncertainty Quantification via Anchor Marginalization. *ArXiv preprint*, abs/2110.02197. (Cited on page 57)

Javier Antorán, James Urquhart Allingham, and José Miguel Hernández-Lobato. 2020. Depth Uncertainty in Neural Networks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. (Cited on page 46)

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. 2018. Understanding Deep Neural Networks with Rectified Linear Units. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. (Cited on pages 98, 102, and 103)

Udit Arora, William Huang, and He He. 2021. Types of Out-of-distribution Texts and how to Detect Them. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10687–10701. (Cited on pages 68, 113, and 116)

Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry P. Vetrov. 2020. Pitfalls of In-domain Uncertainty Estimation and Ensembling in Deep Learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. (Cited on pages 38 and 67)

Jsang Audun. 2018. *Subjective Logic: A Formalism for Reasoning under Uncertainty*. (Cited on page 49)

Razvan Azamfirei, Sapna R. Kudchadkar, and James Fackler. 2023. Large Language Models and the Perils of Their Hallucinations. *Critical Care*, 27(1):1–2. (Cited on page 124)

Joris Baan, Wilker Aziz, Barbara Plank, and Raquel Fernandez. 2022. Stop Measuring Calibration when Humans Disagree. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1892–1915. (Cited on pages 63, 165, 166, and 168)

Joris Baan, Nico Daheim, Evgenia Ilia, Dennis Ulmer, Haau-Sing Li, Raquel Fernández, Barbara Plank, Rico Sennrich, Chrysoula Zerva, and Wilker Aziz. 2023. Uncertainty in Natural Language Generation: From Theory to Applications. *ArXiv preprint*, abs/2307.15703. (Cited on pages 6, 29, 30, 63, 74, 78, 158, 159, 165, and 167)

Joris Baan, Raquel Fernández, Barbara Plank, and Wilker Aziz. 2024. Interpreting Predictive Probabilities: Model Confidence or Human Label Variation? In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 2: Short Papers, St. Julian's, Malta, March 17-22, 2024*, pages 268–277. Association for Computational Linguistics. (Cited on pages 60 and 167)

Annette Baier. 1986. Trust and Antitrust. *Ethics*, 96(2):231–260. (Cited on page 65)

Divya Jyoti Bajpai and Manjesh Kumar Hanawal. 2024. CEEBERT: Cross-Domain Inference in Early Exit BERT. *ArXiv preprint*, abs/2405.15039. (Cited on page 68)

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. 2024. Leak, Cheat, Repeat: Data Contamination and Evaluation Malpractices in Closed-source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 1: Long Papers, St. Julian's, Malta, March 17-22, 2024*, pages 67–93. Association for Computational Linguistics. (Cited on pages 71 and 76)

Neil Band, Xuechen Li, Tengyu Ma, and Tatsunori Hashimoto. 2024. Linguistic Calibration of Language Models. *ArXiv preprint*, abs/2404.00474. (Cited on pages 62 and 67)

Dan Baras. 2023. Carbon Offsetting. *Ethics, Policy & Environment*, pages 1–18. (Cited on page 327)

Rina Foygel Barber, Emmanuel J. Candes, Aaditya Ramdas, and Ryan J. Tibshirani. 2023. Conformal Prediction beyond Exchangeability. *The Annals of Statistics*, 51(2):816–845. (Cited on pages 40, 125, 126, 127, 137, and 139)

Ainhize Barrainkua, Paula Gordaliza, Jose A. Lozano, and Novi Quadrianto. 2024. Uncertainty Matters: Stable Conclusions under Unstable Assessment of Fairness Results. In *International Conference on Artificial Intelligence and Statistics*, pages 1198–1206. PMLR. (Cited on page 67)

Andrew R. Barron. 1994. Approximation and Estimation Bounds for Artificial Neural Networks. *Machine learning*, 14:115–133. (Cited on page 41)

Valerio Basile, Michael Fell, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, Massimo Poesio, and Alexandra Uma. 2021. We Need to Consider Disagreement in Evaluation. In *Proceedings of the 1st Workshop on Benchmarking: Past, Present and Future*, pages 15–21. (Cited on pages 63 and 75)

Elisa Bassignana, Max Müller-Eberstein, Mike Zhang, and Barbara Plank. 2022. Evidence > Intuition: Transferability Estimation for Encoder Selection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 4218–4227. (Cited on page 66)

Elisa Bassignana and Barbara Plank. 2022. CrossRE: A Cross-domain Dataset for Relation Extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3592–3604. (Cited on page 75)

John M. Bates and Clive W. J. Granger. 1969. The Combination of Forecasts. *Journal of the Operational Research Society*, 20(4):451–468. (Cited on page 46)

Eric Bauer and Ron Kohavi. 1999. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine learning*, 36:105–139. (Cited on page 46)

Evan Becker and Stefano Soatto. 2024. Cycles of Thought: Measuring LLM Confidence through Stable Explanations. *ArXiv preprint*, abs/2406.03441. (Cited on page 60)

Anya Belz, Shubham Agarwal, Anastasia Shimorina, and Ehud Reiter. 2021. A Systematic Review of Reproducibility Research in Natural Language Processing. In *Proceedings of the 16th*

Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 381–393. (Cited on pages 71 and 77)

Alessio Benavoli, Giorgio Corani, Janez Demšar, and Marco Zaffalon. 2017. Time for a Change: A Tutorial for Comparing Multiple Classifiers through Bayesian Analysis. *The Journal of Machine Learning Research*, 18(1):2653–2688. (Cited on pages 81 and 95)

Emily M. Bender. 2011. On Achieving and Evaluating Language-independence in NLP. *Linguistic Issues in Language Technology*, 6. (Cited on page 6)

Emily M. Bender and Batya Friedman. 2018. Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science. *Transactions of the Association for Computational Linguistics*, 6:587–604. (Cited on pages 74 and 77)

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be too Big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623. (Cited on pages 71 and 174)

Viktor Bengs, Eyke Hüllermeier, and Willem Waegeman. 2023. On Second-order Scoring Rules for Epistemic Uncertainty Quantification. In *International Conference on Machine Learning*, pages 2078–2091. PMLR. (Cited on page 168)

Martin Benjamin. 2018. Hard Numbers: Language Exclusion in Computational Linguistics and Natural Language Processing. In *Proceedings of the LREC 2018 Workshop "CCURL2018–Sustaining Knowledge Diversity in the Digital Age*, pages 13–18. (Cited on page 75)

Federico Bergamin, Pablo Moreno-Muñoz, Søren Hauberg, and Georgios Arvanitidis. 2024. Riemannian Laplace Approximations for Bayesian Neural Networks. *Advances in Neural Information Processing Systems*, 36. (Cited on page 45)

James O. Berger and Thomas Sellke. 1987. Testing a Point Null Hypothesis: The Irreconcilability of p-values and Evidence. *Journal of the American Statistical Association*, 82(397):112–122. (Cited on page 81)

James Bergstra and Yoshua Bengio. 2012. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13(2). (Cited on pages 78 and 330)

Aadyot Bhatnagar, Huan Wang, Caiming Xiong, and Yu Bai. 2023. Improved Online Conformal Prediction via Strongly Adaptive Online Learning. In *International Conference on Machine Learning*, pages 2337–2363. PMLR. (Cited on page 40)

Umang Bhatt, Javier Antorán, Yunfeng Zhang, Q Vera Liao, Prasanna Sattigeri, Riccardo Fogliato, Gabrielle Melançon, Ranganath Krishnan, Jason Stanley, Omesh Tickoo, et al. 2021. Uncertainty as a Form of Transparency: Measuring, Communicating, and Using Uncertainty. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 401–413. (Cited on page 66)

Joseph R. Biden. 2023. Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence. (Cited on page 175)

Lukas Biewald. 2020. Experiment Tracking with Weights and Biases. Software available from wandb.com. (Cited on pages 326 and 332)

Thomas Bilgram and Britt Keson. 1998. The Construction of a Tagged Danish Corpus. In *Proceedings of the 11th Nordic Conference of Computational Linguistics (NoDaLiDa 1998)*, pages 129–139. (Cited on page 113)

Steven Bird. 2022. Local Languages, Third Spaces, and other High-Resource Scenarios. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7817–7829. (Cited on page 113)

Abeba Birhane, Pratyusha Kalluri, Dallas Card, William Agnew, Ravit Dotan, and Michelle Bao. 2022. The Values Encoded in Machine Learning Research. In *FAccT '22: 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, June 21 - 24, 2022*, pages 173–184. ACM. (Cited on pages 71, 79, and 82)

Bernd Bischl, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, Theresa Ullmann, Marc Becker, Anne-Laure Boulesteix, Difan Deng, and Marius Lindauer. 2023. Hyperparameter Optimization: Foundations, Algorithms, Best Practices, and Open Challenges. *WIREs Data. Mining. Knowl. Discov.*, 13(2). (Cited on pages 77 and 78)

Christopher M. Bishop and Nasser M. Nasrabadi. 2006. *Pattern Recognition and Machine Learning*, volume 4. (Cited on pages 21, 36, 41, 42, and 47)

Txus Blasco, J. Salvador Sánchez, and Vicente García. 2024. A Survey on Uncertainty Quantification in Deep Learning for Financial Time Series Prediction. *Neurocomputing*, 576:127339. (Cited on page 6)

Jarosław Błasiok and Preetum Nakkiran. 2023. Smooth ECE: Principled Reliability Diagrams via Kernel Smoothing. *ArXiv preprint*, abs/2309.12236. (Cited on pages 37 and 151)

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence Estimation for Machine Translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 315–321. (Cited on page 143)

Florian Bley, Sebastian Lapuschkin, Wojciech Samek, and Grégoire Montavon. 2024. Explaining Predictive Uncertainty by Exposing Second-Order Effects. *ArXiv preprint*, abs/2401.17441. (Cited on page 167)

Avrim Blum, Nika Haghtalab, and Ariel D. Procaccia. 2015. Variational Dropout and the Local Reparameterization Trick. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2575–2583. (Cited on page 44)

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight Uncertainty in Neural Network. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1613–1622. (Cited on pages 44, 111, and 330)

Matyas Bohacek and Hany Farid. 2023. Nepotistically Trained Generative-AI Models Collapse. *ArXiv preprint*, abs/2311.12202. (Cited on page 172)

Ondrej Bohdal, Yongxin Yang, and Timothy Hospedales. 2021. Meta-calibration: Learning of Model Calibration Using Differentiable Expected Calibration Error. *ArXiv preprint*, abs/2106.09613. (Cited on page 37)

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214. (Cited on page 129)

Shahin Boluki, Randy Ardywibowo, Siamak Zamani Dadaneh, Mingyuan Zhou, and Xiaoning Qian. 2020. Learnable Bernoulli Dropout for Bayesian Deep Learning. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 3905–3916. (Cited on page 44)

Carlo Bonferroni. 1936. Teoria Statistica delle Classi e Calcolo delle Probabilita. *Pubblicazioni del Instituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8:3–62. (Cited on page 80)

Jean-François Bonnefon and Gaëlle Villejoubert. 2006. Tactful or Doubtful? Expectations of Politeness Explain the Severity Bias in the Interpretation of Probability Phrases. *Psychological Science*, 17(9):747–751. (Cited on page 31)

George Boole. 1854. *An Investigation of the Laws of Thought: On which Are Founded the Mathematical Theories of Logic and Probabilities*, volume 2. (Cited on page 57)

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced Metrics for Measuring Unintended Bias with Real Data for Text Classification. In *Companion Proceedings of the 2019 World Wide Web Conference*, pages 491–500. (Cited on page 96)

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Nazanin Mohammadi Sepahvand, Edward Raff, Kanika Madan, Vikram Voleti, et al. 2021. Accounting for Variance in Machine Learning Benchmarks. *Proceedings of Machine Learning and Systems*, 3. (Cited on pages 71, 76, 80, 83, 84, and 94)

Samuel R. Bowman and George Dahl. 2021. What Will It Take to Fix Benchmarking in Natural Language Understanding? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4843–4855. (Cited on page 75)

John Bradshaw, Alexander G. de G. Matthews, and Zoubin Ghahramani. 2017. Adversarial Examples, Uncertainty, and Transfer Testing Robustness in Gaussian Process Hybrid Deep Networks. *ArXiv preprint*, abs/1707.02476. (Cited on page 47)

Anouck Braggaar and Rob van der Goot. 2021. Challenges in Annotating and Parsing Spoken, Code-switched, Frisian-Dutch Data. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 50–58. (Cited on page 75)

John S. Bridle. 1990. Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition. In *Neurocomputing*, pages 227–236. (Cited on pages 100 and 282)

Glenn W. Brier. 1950. Verification of Forecasts Expressed in Terms of Probability. *Monthly Weather Review*, 78(1):1–3. (Cited on page 151)

Martin Briesch, Dominik Sobania, and Franz Rothlauf. 2023. Large Language Models Suffer from Their own Output: An Analysis of the Self-consuming Training Loop. *ArXiv preprint*, abs/2311.16822. (Cited on page 172)

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. 2011. *Handbook of Markov Chain Monte Carlo.* (Cited on page 81)

Keith Brown. 2005. *Encyclopedia of Language and Linguistics*, volume 1. (Cited on pages 26 and 27)

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of Artificial General Intelligence: Early Experiments with GPT-4. *ArXiv preprint*, abs/2303.12712. (Cited on page 174)

Timothy Campbell. 2021. Offsetting, Denialism, and Risk. (Cited on page 327)

Ricardo J.G.B. Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-based Clustering Based on Hierarchical Density Estimates. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 160–172. Springer. (Cited on pages 147 and 338)

Margarida M. Campos, António Farinhas, Chrysoula Zerva, Mário A. T. Figueiredo, and André F. T. Martins. 2024. Conformal

Prediction for Natural Language Processing: A Survey. *ArXiv preprint*, abs/2405.01976. (Cited on pages 59 and 165)

Michele Caprio, Souradeep Dutta, Kuk Jang, Vivian Lin, Radoslav Ivanov, Oleg Sokolsky, and Insup Lee. 2023. Credal Bayesian Deep Learning. *ArXiv preprint*, abs/2302.09656. (Cited on pages 57 and 58)

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020. With Little Power Comes Great Responsibility. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9263–9274. (Cited on pages 76, 77, and 78)

Sean M. Carroll. 2019. Beyond Falsifiability: Normal Science in a Multiverse. *Why Trust a Theory*, pages 300–314. (Cited on page 72)

Simon Caton and Christian Haas. 2024. Fairness in Machine Learning: A Survey. *ACM Comput. Surv.*, 56(7):166:1–166:38. (Cited on page 3)

Damon Centola, Joshua Becker, Devon Brackbill, and Andrea Baronchelli. 2018. Experimental Evidence for Tipping Points in Social Convention. *Science*, 360(6393):1116–1119. (Cited on page 82)

Sky CH-Wang, Benjamin Van Durme, Jason Eisner, and Chris Kedzie. 2023. Do Androids Know They're only Dreaming of Electric Sheep? (Cited on page 164)

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. 2021. SWAD: Domain Generalization by Seeking Flat Minima. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 22405–22418. (Cited on page 46)

Ilias Chalkidis. 2023. ChatGPT May Pass the Bar Exam Soon, but Has a Long Way to Go for the LexGLUE Benchmark. *ArXiv preprint*, abs/2304.12202. (Cited on page 2)

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019a. Neural Legal Judgment Prediction in English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323. (Cited on page 2)

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019b. Neural Legal Judgment Prediction in English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323. (Cited on page 97)

Bertrand Charpentier, Oliver Borchert, Daniel Zügner, Simon Geisler, and Stephan Günnemann. 2022. Natural Posterior Network: Deep Bayesian Predictive Uncertainty for Exponential Family Distributions. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. (Cited on page 56)

Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. 2020. Posterior Network: Uncertainty Estimation without OOD Samples via Density-based Pseudo-counts. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. (Cited on pages 54, 56, and 279)

Changyou Chen, Nan Ding, Chunyuan Li, Yizhe Zhang, and Lawrence Carin. 2016. Stochastic Gradient MCMC with Stale Gradients. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2937–2945. (Cited on page 43)

Hanjie Chen and Yangfeng Ji. 2022. Explaining Prediction Uncertainty of Pre-trained Language Models by Detecting Uncertain Words in Inputs. *ArXiv preprint*, abs/2201.03742. (Cited on pages 4 and 167)

Jiuhai Chen and Jonas Mueller. 2023. Quantifying Uncertainty in Answers from Any Language Model via Intrinsic and Extrinsic Confidence Assessment. *ArXiv preprint*, abs/2308.16175. (Cited on page 61)

Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. 2023a. Reconcile: Round-table Conference Improves Reasoning via Consensus among Diverse LLMs. *ArXiv preprint*, abs/2309.13007. (Cited on page 62)

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision Transformer: Reinforcement Learning via Sequence Modeling. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 15084–15097. (Cited on page 78)

Wenhu Chen, Yilin Shen, Hongxia Jin, and William Wang. 2018. A Variational Dirichlet Framework for Out-of-distribution Detection. *ArXiv preprint*, abs/1811.07308. (Cited on pages 56 and 281)

Yangyi Chen, Lifan Yuan, Ganqu Cui, Zhiyuan Liu, and Heng Ji. 2023b. A Close Look into the Calibration of Pre-trained Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1343–1367. (Cited on pages 59, 156, and 172)

Muthu Chidambaram, Holden Lee, Colin McSwiggen, and Semon Rezchikov. 2024. How Flawed Is ECE? An Analysis via Logit Smoothing. *ArXiv preprint*, abs/2402.10046. (Cited on page 37)

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4299–4307. (Cited on pages 63 and 339)

Robert T. Clemen. 1989. Combining Forecasts: A Review and Annotated Bibliography. *International journal of forecasting*, 5(4):559–583. (Cited on page 46)

climeworks. 2022. Climeworks. https://climeworks.com/. Accessed: 2022-06-22. (Cited on page 327)

Adam D. Cobb and Brian Jalaian. 2021. Scaling Hamiltonian Monte Carlo Inference for Bayesian Neural Networks with Symmetric Splitting. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021*, volume 161 of *Proceedings of Machine Learning Research*, pages 675–685. (Cited on page 42)

William G. Cochran. 1934. The Distribution of Quadratic Forms in a Normal System, with Applications to the Analysis of Covariance. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 30, pages 178–191. Cambridge University Press. (Cited on page 16)

Arman Cohan, Sydney Young, and Nazli Goharian. 2016. Triaging Mental Health Forum Posts. In *Proceedings of the Third Workshop on Computational Linguistics and Clinical Psychology*, pages 143–147. (Cited on page 2)

K. Bretonnel Cohen, Jingbo Xia, Pierre Zweigenbaum, Tiffany Callahan, Orin Hargraves, Foster Goss, Nancy Ide, Aurélie Névéol, Cyril Grouin, and Lawrence E. Hunter. 2018. Three Dimensions of Reproducibility in Natural Language Processing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. (Cited on page 72)

A. Feder Cooper, Katherine Lee, Madinha Zahrah Choksi, Barocas Solon, Christopher De Sa, James Grimmelmann, Jon Kleinberg, Siddhartha Sen, and Baobao Zhang. 2024. Is My Prediction Arbitrary? The Confounding Effects of Variance in Fair Classification Benchmarks. (Cited on page 67)

Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. 2019. Addressing Failure Prediction by Learning Model Confidence. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2898–2909. (Cited on page 57)

Charles Corbière, Nicolas Thome, Antoine Saporta, Tuan-Hung Vu, Matthieu Cord, and Patrick Perez. 2021. Confidence Estimation via Auxiliary Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6043–6055. (Cited on page 57)

Francesco Croce and Matthias Hein. 2018. A Randomized Gradient-free Attack on ReLU Networks. In *German Conference on Pattern Recognition*, pages 215–227. Springer. (Cited on pages 103 and 104)

Alicia Curth, Patrick Thoral, Wilco van den Wildenberg, Peter Bijlstra, Daan de Bruin, Paul W. G. Elbers, and Mattia Fornasa. 2019. Transferring Clinical Prediction Models across Hospitals and Electronic Health Record Systems. In *Machine Learning and Knowledge Discovery in Databases - International Workshops of ECML PKDD 2019, Würzburg, Germany, September 16-20, 2019, Proceedings, Part I*, volume 1167 of *Communications in Computer and Information Science*, pages 605–621. (Cited on page 99)

Andreas C. Damianou and Neil D. Lawrence. 2013. Deep Gaussian Processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2013, Scottsdale, AZ, USA, April 29 - May 1, 2013*, volume 31 of *JMLR Workshop and Conference Proceedings*, pages 207–215. (Cited on page 47)

Alexander D'Amour, Katherine A. Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan Karthikesalingam, Mario Lucic, Yi-An Ma, Cory Y. McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Natarajan, Christopher Nielson, Thomas F. Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Martin Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov, Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D. Sculley. 2022. Underspecification Presents Challenges for Credibility in Modern Machine Learning. *Journal of Machine Learning Research (JMLR)*, 23:226:1–226:61. (Cited on pages 7, 68, 74, and 111)

Soham Dan and Dan Roth. 2021. On the Effects of Transformer Size on In- and Out-of-domain Calibration. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2096–2101. (Cited on pages 59, 116, and 172)

Francesco D'Angelo and Vincent Fortuin. 2021. Repulsive Deep Ensembles Are Bayesian. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 3451–3465. (Cited on page 46)

Kahneman Daniel. 2017. *Thinking, Fast and Slow.* (Cited on pages 66 and 68)

Constantinos Daskalakis, Petros Dellaportas, and Aristeidis Panos. 2020. Scalable Gaussian Processes, with Guarantees: Kernel Approximations and Deep Feature Extraction. (Cited on page 47)

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with Disagreements: Looking beyond the Majority Vote in Subjective Annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110. (Cited on page 63)

Erik Daxberger and José Miguel Hernández-Lobato. 2019. Bayesian Variational Autoencoders for Unsupervised Out-of-distribution Detection. *ArXiv preprint*, abs/1912.05651. (Cited on page 166)

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. 2021a. Laplace Redux - Effortless Bayesian Deep Learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021,* *December 6-14, 2021, virtual*, pages 20089–20103. (Cited on page 45)

Erik A. Daxberger, Eric T. Nalisnick, James Urquhart Allingham, Javier Antorán, and José Miguel Hernández-Lobato. 2021b. Bayesian Deep Learning via Subnetwork Inference. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 2510–2521. (Cited on page 45)

Marco De Angelis and Ander Gray. 2021. Why the 1-Wasserstein Distance Is the Area between the two Marginal CDFs. *ArXiv preprint*, abs/2111.03570. (Cited on page 86)

Mostafa Dehghani, Yi Tay, Alexey A Gritsenko, Zhe Zhao, Neil Houlsby, Fernando Diaz, Donald Metzler, and Oriol Vinyals. 2021. The Benchmark Lottery. (Cited on page 76)

Frederik Michel Dekking, Cornelis Kraaikamp, Hendrik Paul Lopuhaä, and Ludolf Erwin Meester. 2005. *A Modern Introduction to Probability and Statistics: Understanding why and how*, volume 488. (Cited on page 14)

Eustasio del Barrio, Juan A. Cuesta-Albertos, and Carlos Matrán. 2018a. An Optimal Transportation Approach for Assessing Almost Stochastic Order. In *The Mathematics of the Uncertain*, pages 33–44. (Cited on pages 71, 85, 86, 114, 136, 137, and 151)

Eustasio del Barrio, Juan A. Cuesta-Albertos, and Carlos Matrán. 2018b. Some Indices to Measure Departures from Stochastic Order. *ArXiv preprint*, abs/1804.02905. (Cited on page 85)

Arthur P. Dempster. 1968. A Generalization of Bayesian Inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, 30(2):205–232. (Cited on page 49)

Janez Demšar. 2008. On the Appropriateness of Statistical Tests in Machine Learning. In *Workshop on Evaluation Methods for Machine Learning in conjunction with ICML*, page 65. Citeseer. (Cited on page 81)

Naihao Deng, Xinliang Frederick Zhang, Siyang Liu, Winston Wu, Lu Wang, and Rada Mihalcea. 2023. You Are What You Annotate: Towards Better Models through Annotator Representations. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 12475–12498. Association for Computational Linguistics. (Cited on pages 64 and 75)

Zhijie Deng, Feng Zhou, Jianfei Chen, Guoqiang Wu, and Jun Zhu. 2022. Deep Ensemble as a Gaussian Process Approximate Posterior. *ArXiv preprint*, abs/2205.00163. (Cited on page 46)

Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. 2018. Decomposition of Uncertainty in Bayesian Deep Learning for Efficient and Risk-sensitive Learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1192–1201. (Cited on pages 48 and 101)

Armen Der Kiureghian and Ove Ditlevsen. 2009. Aleatory or Epistemic? Does It Matter? *Structural safety*, 31(2):105–112. (Cited on page 167)

Shrey Desai and Greg Durrett. 2020. Calibration of Pre-trained Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302. (Cited on pages 59, 116, and 156)

Gianluca Detommaso, Martin Bertran, Riccardo Fogliato, and Aaron Roth. 2024. Multicalibration for Confidence Scoring in LLMs. *ArXiv preprint*, abs/2404.04689. (Cited on pages 59 and 67)

Nicolas Deutschmann, Marvin Alberts, and María Rodríguez Martínez. 2024. Conformal Autoregressive Generation: Beam Search with Coverage Guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11775–11783. (Cited on page 59)

Neşe Devenot. 2023. Tescreal Hallucinations: Psychedelic and AI Hype as Inequality Engines. *Journal of Psychedelic Studies*, 7(S1):22–39. (Cited on page 171)

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. (Cited on pages 5, 63, 78, 114, 173, 174, 333, and 337)

Filip Devos. 2003. Semantic Vagueness And Lexical Polyvalence. *Studia Linguistica*, 57(3):121–141. (Cited on page 26)

Terrance DeVries and Graham W. Taylor. 2018. Learning Confidence for Out-of-distribution Detection in Neural Networks. *ArXiv preprint*, abs/1802.04865. (Cited on page 68)

Shehzaad Dhuliawala, Vilém Zouhar, Mennatallah El-Assady, and Mrinmaya Sachan. 2023. A Diachronic Perspective on User Trust in AI Under Uncertainty. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5567–5580. (Cited on pages 66, 140, and 165)

Thomas G. Dietterich. 2000. Ensemble Methods in Machine Learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer. (Cited on page 46)

Thomas G. Dietterich and Alex Guyer. 2022. The Familiarity Hypothesis: Explaining the Behavior of Deep Open Set Methods. *Pattern Recognition*, 132:108931. (Cited on page 161)

Jesse Dodge and Noah A. Smith. 2020. Reproducibility at EMNLP 2020. `https://2020.emnlp.org/blog/2020-05-20-reproducibility`. (Cited on page 72)

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image Is Worth 16X16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021.* (Cited on page 78)

Rotem Dror, Gili Baumer, Marina Bogomolov, and Roi Reichart. 2017. Replicability Analysis for Natural Language Processing: Testing Significance with Multiple Datasets. *Transactions of the Association for Computational Linguistics*, 5:471–486. (Cited on page 80)

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392. (Cited on pages 80 and 83)

Rotem Dror, Lotem Peled-Cohen, Segev Shlomov, and Roi Reichart. 2020. Statistical Significance Testing for Natural Language Processing. *Synthesis Lectures on Human Language Technologies*, 13(2):1–116. (Cited on page 83)

Rotem Dror, Segev Shlomov, and Roi Reichart. 2019. Deep Dominance - How to Properly Compare Deep Neural Models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2785. (Cited on pages 71, 80, 85, 86, 114, 136, 137, 151, 328, and 329)

Chris Drummond. 2009. Replicability Is not Reproducibility: Nor Is It Good Science. *Proceedings of the Evaluation Methods for Machine Learning Workshop at the 26th ICML.* (Cited on page 72)

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. 2023. Shortcut Learning of Large Language Models in Natural Language Understanding. *Communications of the ACM*, 67(1):110–120. (Cited on page 156)

Du, Wenda, and Han, Qing. 2021. Research on Application of Artificial Intelligence in the Movie Industry. volume 12076. (Cited on page 170)

Hanyu Duan, Yi Yang, and Kar Yan Tam. 2024. Do LLMs Know about Hallucination? An Empirical Investigation of LLM's Hidden States. *ArXiv preprint*, abs/2402.09733. (Cited on page 164)

Matthew M. Dunlop, Mark A. Girolami, Andrew M. Stuart, and Aretha L. Teckentrup. 2018. How Deep Are Deep Gaussian Processes? *Journal of Machine Learning Research*, 19(54):1–46. (Cited on page 47)

Nikita Durasov, Timur M. Bagautdinov, Pierre Baqué, and Pascal Fua. 2021. Masksembles for Uncertainty Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 13539–13548. (Cited on pages 44 and 46)

Michael Dusenberry, Ghassen Jerfel, Yeming Wen, Yi-An Ma, Jasper Snoek, Katherine A. Heller, Balaji Lakshminarayanan, and Dustin Tran. 2020. Efficient and Scalable Bayesian Neural Nets with Rank-1 Factors. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 2782–2792. (Cited on page 46)

Vincent Dutordoir, James Hensman, Mark van der Wilk, Carl Henrik Ek, Zoubin Ghahramani, and Nicolas Durrande. 2021. Deep Neural Networks as Point Estimates for Deep Gaussian Processes. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021,* *NeurIPS 2021, December 6-14, 2021, virtual*, pages 9443–9455. (Cited on page 47)

Sayna Ebrahimi, William Gan, Dian Chen, Giscard Biamby, Kamyar Salahi, Michael Laielli, Shizhan Zhu, and Trevor Darrell. 2020. Minimax Active Learning. *ArXiv preprint*, abs/2012.10467. (Cited on page 69)

Bradley Efron. 1992. Bootstrap Methods: Another Look at the Jackknife. In *Breakthroughs in statistics: Methodology and distribution*, pages 569–593. (Cited on page 17)

Bradley Efron. 2022. *Exponential Families in Theory and Practice.* (Cited on page 21)

Bradley Efron and Robert J. Tibshirani. 1994. *An Introduction to the Bootstrap.* (Cited on pages 87 and 151)

Bryan Eikema. 2024. The Effect of Generalisation on the Inadequacy of the Mode. In *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024)*, pages 87–92. (Cited on pages 64, 123, and 162)

Bryan Eikema and Wilker Aziz. 2020. Is MAP Decoding all You Need? The Inadequacy of the Mode in Neural Machine Translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520. (Cited on pages 64, 123, 138, and 162)

Yousef El-Laham, Niccolò Dalmasso, Elizabeth Fons, and Svitlana Vyetrenko. 2023. Deep Gaussian Mixture Ensembles. In *Uncertainty in Artificial Intelligence*, pages 549–559. PMLR. (Cited on page 46)

Paul E Engelhardt and Fernanda Ferreira. 2010. Processing Coordination Ambiguity. *Language and speech*, 53(4):494–509. (Cited on page 28)

Piero Esposito. 2020. Blitz - Bayesian Layers in Torch Zoo (a Bayesian Deep Learing Library for Torch). https://github.com/piEsposito/blitz-bayesian-deep-learning/. (Cited on page 333)

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, pages 226–231. (Cited on pages 147 and 338)

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, et al. 2024. Fact-checking the Output of Large Language Models via Token-level Uncertainty Quantification. *ArXiv preprint*, abs/2403.04696. (Cited on page 60)

Wade Fagen-Ulmschneider. 2015. Perception of Probability Words. https://waf.cs.illinois.edu/visualizations/Perception-of-Probability-Words/. Accessed: 2024-06-10. (Cited on page 146)

Ahmad Faiz, Sotaro Kaneda, Ruhan Wang, Rita Chukwunyere Osi, Prateek Sharma, Fan Chen, and Lei Jiang. 2023. LLMCarbon: Modeling the End-to-end Carbon Footprint of Large Language Models. In *The Twelfth International Conference on Learning Representations*. (Cited on page 326)

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. 2021. Beyond English-centric Multilingual Machine Translation. *Journal of Machine Learning Research (JMLR)*, 22:107:1–107:48. (Cited on page 129)

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898. (Cited on pages 92, 123, and 135)

António Farinhas, Chrysoula Zerva, Dennis Ulmer, and André F. T. Martins. 2024. Non-exchangeable Conformal Risk Control. In *The Twelfth International Conference on Learning Representations*. (Cited on pages 40, 127, 138, and 164)

Yassir Fathullah and Mark J. F. Gales. 2022. Self-distribution Distillation: Efficient Uncertainty Estimation. In *Uncertainty in Artificial Intelligence, Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence, UAI 2022, 1-5 August 2022, Eindhoven, The Netherlands*, volume 180 of *Proceedings of Machine Learning Research*, pages 663–673. (Cited on page 55)

Yassir Fathullah, Puria Radmard, Adian Liusie, and Mark J. F. Gales. 2024. Efficient Estimation of Sequence-level Attributes with Proxies. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 1: Long Papers, St. Julian's, Malta, March 17-22, 2024*, pages 1478–1496. Association for Computational Linguistics. (Cited on pages 57 and 168)

Marco Federici, Ryota Tomioka, and Patrick Forré. 2021. An Information-theoretic Approach to Distribution Shifts. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 17628–17641. (Cited on page 113)

Zhengcong Fei, Xu Yan, Shuhui Wang, and Qi Tian. 2022. DeeCap: Dynamic Early Exiting for Efficient Image Captioning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 12206–12216. (Cited on page 68)

Shai Feldman, Stephen Bates, and Yaniv Romano. 2021. Improving Conditional Coverage via Orthogonal Quantile Regression. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 2060–2071. (Cited on page 166)

Yunzhen Feng, Elvis Dohmatob, Pu Yang, Francois Charton, and Julia Kempe. A Tale of Tails: Model Collapse as a Change of Scaling Laws. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models.* (Cited on page 172)

Patrick Fernandes, António Farinhas, Ricardo Rei, José G. C. de Souza, Perez Ogayo, Graham Neubig, and Andre Martins. 2022. Quality-aware Decoding for Neural Machine Translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1396–1412. (Cited on page 138)

Javier Ferrando, Gerard I. Gállego, Belen Alastruey, Carlos Escolano, and Marta R. Costa-jussà. 2022. Towards Opening the Black Box of Neural Machine Translation: Source and Target Interpretations of the Transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8756–8769. (Cited on page 164)

Fernanda Ferreira and John M. Henderson. 1991. Recovery From Misanalyses Of Garden-Path Sentences. *Journal of Memory and Language*, 30(6):725–745. (Cited on pages 4 and 29)

Adam Fisch, Tal Schuster, Tommi S. Jaakkola, and Regina Barzilay. 2022. Conformal Prediction Sets with Limited False Positives. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 6514–6532. (Cited on page 40)

Jerry Fodor. 2001. Language, Thought and Compositionality. *Royal Institute of Philosophy Supplements*, 48:227–242. (Cited on page 27)

Antske Fokkens, Marieke van Erp, Marten Postma, Ted Pedersen, Piek Vossen, and Nuno Freire. 2013. Offspring from Reproduction Problems: What Replication Failure Teaches Us. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1691–1701. (Cited on pages 72, 75, and 77)

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. 2019. Deep Ensembles: A Loss Landscape Perspective. *ArXiv preprint*, abs/1912.02757. (Cited on page 46)

Vincent Fortuin, Adrià Garriga-Alonso, Sebastian W. Ober, Florian Wenzel, Gunnar Rätsch, Richard E. Turner, Mark van der Wilk, and Laurence Aitchison. 2022. Bayesian Neural Network Priors Revisited. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. (Cited on page 38)

Meire Fortunato, Charles Blundell, and Oriol Vinyals. 2017. Bayesian Recurrent Neural Networks. *ArXiv preprint*, abs/1704.02798. (Cited on page 111)

Rina Foygel Barber, Emmanuel J. Candes, Aaditya Ramdas, and Ryan J. Tibshirani. 2021. The Limits of Distribution-free Conditional Predictive Inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482. (Cited on page 166)

Gianni Franchi, Xuanlong Yu, Andrei Bursuc, Emanuel Aldea, Severine Dubuisson, and David Filliat. 2022. Latent Discriminant Deterministic Uncertainty. In *European Conference on Computer Vision*, pages 243–260. Springer. (Cited on page 57)

Bruce Fraser. 1975. Hedged Performatives. In *Speech Acts*, pages 187–210. (Cited on page 30)

Lyn Frazier, Alan Munn, and Charles Clifton. 2000. Processing Coordinate Structures. *Journal of Psycholinguistic Research*, 29:343–370. (Cited on page 28)

Linton C. Freeman. 1965. Elementary Applied Statistics: For Students in Behavioral Science. (Cited on page 47)

Markus Freitag, Behrooz Ghorbani, and Patrick Fernandes. 2023. Epsilon Sampling Rocks: Investigating Sampling Strategies for Minimum Bayes Risk Decoding for Machine Translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 9198–9209. Association for Computational Linguistics. (Cited on page 138)

Steven Frisson. 2009. Semantic Underspecification in Language Processing. *Language and Linguistics Compass*, 3(1):111–127. (Cited on pages 26 and 27)

Yarin Gal. 2016. Uncertainty in Deep Learning. (Cited on pages 41 and 47)

Yarin Gal and Zoubin Ghahramani. 2016a. A Theoretically-grounded Application of Dropout in Recurrent Neural Networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1019–1027. (Cited on pages 44, 111, and 332)

Yarin Gal and Zoubin Ghahramani. 2016b. Dropout as a Bayesian Approximation: Representing Model Uncertainty on Deep Learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1050–1059. (Cited on pages 44, 50, 109, and 111)

Yarin Gal, Jiri Hron, and Alex Kendall. 2017a. Concrete Dropout. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3581–3590. (Cited on page 44)

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017b. Deep Bayesian Active Learning with Image Data. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1183–1192. (Cited on page 69)

Mirta Galesic and Rocio Garcia-Retamero. 2010. Statistical Numeracy for Health: A Cross-Cultural Comparison with Probabilistic National Samples. *Archives of internal medicine*, 170(5):462–468. (Cited on page 66)

Bolin Gao and Lacra Pavel. 2017. On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning. *ArXiv preprint*, abs/1704.00805. (Cited on page 284)

Songyang Gao, Qiming Ge, Wei Shen, Shihan Dou, Junjie Ye, Xiao Wang, Rui Zheng, Yicheng Zou, Zhi Chen, Hang Yan, et al. 2024a. Linear Alignment: A Closed-form Solution for Aligning Human Preferences without Tuning and Feedback. *ArXiv preprint*, abs/2401.11458. (Cited on pages 138 and 164)

Xiang Gao, Jiaxin Zhang, Lalla Mouatadid, and Kamalika Das. 2024b. SPUQ: Perturbation-Based Uncertainty Quantification for Large Language Models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 1: Long Papers, St. Julian's, Malta, March 17-22, 2024*, pages 2336–2346. Association for Computational Linguistics. (Cited on page 61)

Carlos García Rodríguez, Jordi Vitrià, and Oscar Mora. 2020. Uncertainty-based Human-in-the-loop Deep Learning for Land Cover Segmentation. *Remote Sensing*, 12(22):3836. (Cited on page 69)

Jacob R. Gardner, Geoff Pleiss, Kilian Q. Weinberger, David Bindel, and Andrew Gordon Wilson. 2018. GPyTorch: Blackbox Matrix-matrix Gaussian Process Inference with GPU Acceleration. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7587–7597. (Cited on page 333)

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P. Vetrov, and Andrew Gordon Wilson. 2018. Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8803–8812. (Cited on page 46)

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna M. Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for Datasets. *Commun. ACM*, 64(12):86–92. (Cited on page 74)

Dirk Geeraerts. 1993. Vagueness's Puzzles, Polysemy's Vagaries. (Cited on page 27)

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2022. Repairing the Cracked Foundation: A Survey of Obstacles in Evaluation Practices for Generated Text. *ArXiv preprint*, abs/2202.06935. (Cited on pages 71 and 78)

Yonatan Geifman and Ran El-Yaniv. 2019. Selectivenet: A Deep Neural Network with an Integrated Reject Option. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2151–2159. (Cited on page 57)

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, Donald B. Rubin, John Carlin, Hal Stern, Donald Rubin, and David Dunson. 2021. Bayesian Data Analysis Third Edition. (Cited on pages 21, 42, 81, 84, and 95)

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2023. A Survey of Language Model Confidence Estimation and Calibration. *ArXiv preprint*, abs/2311.08298. (Cited on pages 59 and 165)

Walter Gerych, Yara Rizk, Vatche Isahagian, Vinod Muthusamy, Evelyn Duesterwald, and Praveen Venkateswaran. 2024. Who Knows the Answer? Finding the Best Model and Prompt for each Query Using Confidence-based Search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18065–18072. (Cited on page 68)

Arindam Ghosh, Thomas Schaaf, and Matthew Gormley. 2022. AdaFocal: Calibration-aware Adaptive Focal Loss. *Advances in Neural Information Processing Systems*, 35:1583–1595. (Cited on page 37)

Isaac Gibbs and Emmanuel J. Candès. 2021. Adaptive Conformal Inference under Distribution Shift. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 1660–1672. (Cited on page 40)

Isaac Gibbs, John J. Cherian, and Emmanuel J. Candès. 2023. Conformal Prediction with Conditional Guarantees. *ArXiv preprint*, abs/2305.12616. (Cited on page 166)

Eric W. Gibson. 2021. The Role of p-values in Judging the Strength of Evidence and Realistic Replication Expectations. *Statistics in Biopharmaceutical Research*, 13(1):6–18. (Cited on page 95)

Alexios Gidiotis and Grigorios Tsoumakas. 2022. Should We Trust This Summary? Bayesian Abstractive Summarization to the Rescue. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4119–4131. (Cited on page 60)

Mario Giulianelli, Joris Baan, Wilker Aziz, Raquel Fernández, and Barbara Plank. 2023. What Comes Next? Evaluating Uncertainty in Neural Text Generators against Human Production Variability. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 14349–14371. Association for Computational Linguistics. (Cited on pages 64 and 167)

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep Sparse Rectifier Neural Networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323. PJMLR Workshop and Conference Proceedings. (Cited on page 100)

Taisiya Glushkova, Chrysoula Zerva, Ricardo Rei, and André F. T. Martins. 2021. Uncertainty-aware Machine Translation Evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3920–3938. (Cited on page 143)

Tilmann Gneiting and Adrian E. Raftery. 2007. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378. (Cited on page 37)

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 2019. OpenWebText Corpus. `http://Skylion007.github.io/OpenWebTextCorpus`. (Cited on page 129)

Gold Standard. 2024. The Gold Standard Marketplace. Last accessed 21.06.24. (Cited on page 327)

Zhiqiang Gong, Ping Zhong, and Weidong Hu. 2019. Diversity in Machine Learning. *IEEE Access*, 7:64323–64350. (Cited on page 75)

Irving John Good. 1971. 46656 Varieties of Bayesians. *American Statistician*, 25(5):62. (Cited on page 19)

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press. (Cited on pages xiv and 33)

Kyle Gorman and Steven Bedrick. 2019. We Need to Talk about Standard Splits. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2786–2791. (Cited on pages 71 and 76)

Johannes Graën, Mara Bertamini, Martin Volk, Mark Cieliebak, Don Tuggener, and Fernando Benites. 2018. Cutter–A Universal Multilingual Tokenizer. In *CEUR Workshop Proceedings*, 2226, pages 75–81. CEUR-WS. (Cited on page 163)

Alex Graves. 2011. Practical Variational Inference for Neural Networks. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 2348–2356. (Cited on page 43)

Alex Graves. 2012. Sequence Transduction with Recurrent Neural Networks. *arXiv preprint arXiv:1211.3711.* (Cited on page 135)

Sander Greenland, Stephen J. Senn, Kenneth J. Rothman, John B. Carlin, Charles Poole, Steven N. Goodman, and Douglas G. Altman. 2016. Statistical Tests, p-values, Confidence Intervals, and Power: A Guide to Misinterpretations. *European journal of epidemiology*, 31(4):337–350. (Cited on pages 80 and 95)

H. Paul Grice. 1957. Meaning. *The Philosophical Review*, 66(3):377–388. (Cited on page 30)

Stefan Th. Gries. 2015. Polysemy. *Handbook of cognitive linguistics*, 39:472–490. (Cited on page 26)

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. 2024. OLmo: Accelerating the Science of Language Models. *ArXiv preprint*, abs/2402.00838. (Cited on pages 78, 90, and 174)

Cornelia Gruber, Katharina Hechinger, Matthias Aßenmacher, Göran Kauermann, and Barbara Plank. 2024. More Labels or Cases? Assessing Label Variation in Natural Language Inference. In *Proceedings of the Third Workshop on Understanding Implicit and Underspecified Language*, pages 22–32. (Cited on pages 63, 64, 75, and 166)

Cornelia Gruber, Patrick Oliver Schenk, Malte Schierholz, Frauke Kreuter, and Göran Kauermann. 2023. Sources of Uncertainty in Machine Learning–A Statisticians' View. *ArXiv preprint*, abs/2305.16703. (Cited on page 167)

Sebastian Gruber and Florian Buettner. 2022. Better Uncertainty Calibration via Proper Scores for Classification and Beyond. *Advances in Neural Information Processing Systems*, 35:8618–8632. (Cited on page 37)

Nuno M. Guerreiro, Duarte M. Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André F.T. Martins. 2023a. Hallucinations in Large Multilingual Translation Models. *Transactions of the Association for Computational Linguistics*, 11:1500–1517. (Cited on page 164)

Nuno M. Guerreiro, Elena Voita, and André F.T. Martins. 2023b. Looking for a Needle in a Haystack: A Comprehensive Study of Hallucinations in Neural Machine Translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1059–1075. (Cited on page 123)

Yu Gui, Ying Jin, and Zhimei Ren. 2024. Conformal Alignment: Knowing when to Trust Foundation Models with Guarantees. *ArXiv preprint*, abs/2405.10301. (Cited on pages 59 and 164)

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. 2023. Reinforced Self-training (ReST) for Language Modeling. *ArXiv preprint*, abs/2308.08998. (Cited on page 172)

Odd Erik Gundersen and Sigbjørn Kjensmo. 2018. State of the Art: Reproducibility in Artificial Intelligence. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1644–1651. (Cited on page 71)

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. (Cited on pages 37, 98, 115, and 130)

Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. 2023. The Curious Decline of Linguistic Diversity: Training Language Models on Synthetic Text. *ArXiv preprint*, abs/2311.09807. (Cited on page 172)

Kartik Gupta, Amir Rahimi, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard Hartley. 2021. Calibration of Neural Networks Using Splines. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021.* (Cited on page 37)

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. 2024. Language Model Cascades: Token-level Uncertainty and Beyond. *ArXiv preprint*, abs/2404.10136. (Cited on pages 60 and 68)

Sangchul Hahn and Heeyoul Choi. 2019. Self-knowledge Distillation in Natural Language Processing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 423–430. (Cited on page 133)

Lars Kai Hansen and Peter Salamon. 1990. Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001. (Cited on page 46)

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array Programming with NumPy. *Nature*, 585(7825):357–362. (Cited on page 325)

Manuel Haussmann, Sebastian Gerwinn, and Melih Kandemir. 2019. Bayesian Evidential Deep Learning with PAC Regularization. *ArXiv preprint*, abs/1906.00816. (Cited on pages 49 and 55)

Katri Haverinen, Jenna Nyblom, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Anna Missilä, Stina Ojala, Tapio Salakoski, and Filip Ginter. 2014. Building the Essential Resources for Finnish: The Turku Dependency Treebank. *Language Resources and Evaluation*, 48:493–531. Open access. (Cited on pages 112 and 113)

Jianfeng He. 2024. Uncertainty Estimation on Natural Language Processing. (Cited on page 165)

Jianfeng He, Linlin Yu, Shuo Lei, Chang-Tien Lu, and Feng Chen. 2023a. Uncertainty Estimation on Sequential Labeling via Uncertainty Transmission. *ArXiv preprint*, abs/2311.08726. (Cited on page 59)

Jianxing He, Sally L. Baxter, Jie Xu, Jiming Xu, Xingtao Zhou, and Kang Zhang. 2019. The Practical Implementation of Artificial Intelligence Technologies in Medicine. *Nature medicine*, 25(1):30–36. (Cited on page 74)

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2021a. Efficient Nearest Neighbor Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5703–5714. (Cited on page 124)

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023b. DeBERTaV3: Improving DeBERTa Using ElectrasStyle Pre-training with Gradient-disentangled Embedding Sharing. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. (Cited on page 147)

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. (Cited on page 147)

Ramya Hebbalaguppe, Jatin Prakash, Neelabh Madan, and Chetan Arora. 2022. A Stitch in Time Saves Nine: A Train-time Regularizing Loss for Improved Neural Network Calibration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 16060–16069. (Cited on page 37)

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A Survey on Recent Approaches for Natural Language Processing in Low-resource Scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568. (Cited on page 113)

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. 2019. Why ReLU Networks Yield High-confidence Predictions Far Away from the Training Data and How to Mitigate the Problem. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 41–50. (Cited on pages 68, 98, 103, 104, 105, 106, 112, and 283)

Patrick Hemmer, Niklas Kühl, and Jakob Schöffer. 2022. DEAL: Deep Evidential Active Learning for Image Classification. *Deep Learning Applications, Volume 3*, pages 171–192. (Cited on page 69)

Roee Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 9318–9333. Association for Computational Linguistics. (Cited on page 91)

Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. 2020. Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning. *Journal of Machine Learning Research*, 21(248):1–43. (Cited on page 71)

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2018. Deep Reinforcement Learning that Matters. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3207–3214. (Cited on page 79)

Dan Hendrycks and Thomas G. Dietterich. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. (Cited on page 133)

Dan Hendrycks and Kevin Gimpel. 2017. A Baseline for Detecting Misclassified and Out-of-distribution Examples in Neural Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. (Cited on pages 40, 54, 100, and 109)

Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. 2020. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. (Cited on page 38)

James Hensman and Neil D. Lawrence. 2014. Nested Variational Compression in Deep Gaussian Processes. *arXiv preprint arXiv:1412.1370*. (Cited on page 47)

José Miguel Hernández-Lobato and Ryan P. Adams. 2015. Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. In *Proceedings of the 32nd International Conference* *on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1861–1869. (Cited on page 44)

Daniel Hershcovich, Stella Frank, Heather Lent, Miryam de Lhoneux, Mostafa Abdou, Stephanie Brandl, Emanuele Bugliarello, Laura Cabello Piqueras, Ilias Chalkidis, Ruixiang Cui, Constanza Fierro, Katerina Margatina, Phillip Rust, and Anders Søgaard. 2022. Challenges and Strategies in Cross-cultural NLP. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6997–7013. (Cited on page 6)

Donald Hindle and Mats Rooth. 1990. Structural Ambiguity and Lexical Relations. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990.* (Cited on page 28)

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the Knowledge in a Neural Network. *ArXiv preprint*, abs/1503.02531. (Cited on page 55)

Geoffrey E. Hinton and Drew Van Camp. 1993. Keeping the Neural Networks Simple by Minimizing the Description Length of the Weights. In *Proceedings of the Sixth Annual Conference on Computational learning Theory*, pages 5–13. (Cited on page 43)

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural computation*, 9(8):1735–1780. (Cited on pages 111 and 338)

Paul Hofman, Yusuf Sale, and Eyke Hüllermeier. 2024. Quantifying Aleatoric and Epistemic Uncertainty with Proper Scoring Rules. *ArXiv preprint*, abs/2404.12215. (Cited on page 168)

Andreas Nugaard Holm, Dustin Wright, and Isabelle Augenstein. 2023. Revisiting Softmax for Uncertainty Approximation in Text Classification. *Information*, 14(7):420. (Cited on page 59)

Janet Holmes. 1982. Expressing Doubt and Certainty in English. *RELC journal*, 13(2):9–28. (Cited on page 30)

Benedikt Höltgen and Robert C. Williamson. 2023. On the Richness of Calibration. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 1124–1138. (Cited on page 147)

Thomas Holtgraves and Audrey Perdew. 2016. Politeness and the Communication Of Uncertainty. *Cognition*, 154:1–10. (Cited on page 30)

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. (Cited on pages 92, 123, 130, and 162)

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to Write with Cooperative Discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1638–1649. (Cited on pages 92 and 135)

Sara Hooker. 2021. The Hardware Lottery. *Communications of the ACM*, 64(12):58–65. (Cited on page 82)

Stephen C. Hora. 1996. Aleatory and Epistemic Uncertainty in Probability Elicitation with an Example from Hazardous Waste Management. *Reliability Engineering & System Safety*, 54(2-3):217–223. (Cited on page 167)

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multi-layer Feedforward Networks are Universal Approximators. *Neural networks*, 2(5):359–366. (Cited on page 41)

Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. 2023a. Decomposing Uncertainty for Large Language Models through Input Clarification Ensembling. *ArXiv preprint*, abs/2311.08718. (Cited on page 61)

Bairu Hou, Joe O'Connor, Jacob Andreas, Shiyu Chang, and Yang Zhang. 2023b. Promptboosting: Black-box Text Classification with ten Forward-passes. In *International Conference on Machine Learning*, pages 13309–13324. PMLR. (Cited on page 61)

Dirk Hovy and Shrimai Prabhumoye. 2021. Five Sources of Bias in Natural Language Processing. *Language and Linguistics Compass*, 15(8):e12432. (Cited on pages 74 and 82)

Dirk Hovy and Shannon L. Spruit. 2016. The Social Impact of Natural Language Processing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 591–598. (Cited on pages 71 and 170)

Hengtong Hu, Lingxi Xie, Xinyue Huo, Richang Hong, and Qi Tian. 2022. Vibration-based Uncertainty Estimation for Learning from Limited Supervision. In *European Conference on Computer Vision*, pages 160–176. Springer. (Cited on page 56)

Jennifer Hu, Roger Levy, Judith Degen, and Sebastian Schuster. 2023a. Expectations over Unspoken Alternatives Predict Pragmatic Inferences. *Transactions of the Association for Computational Linguistics*, 11:885–901. (Cited on page 167)

Mengting Hu, Zhen Zhang, Shiwan Zhao, Minlie Huang, and Bingzhe Wu. 2023b. Uncertainty in Natural Language Processing: Sources, Quantification, and Applications. *ArXiv preprint*, abs/2306.04459. (Cited on page 165)

Yibo Hu, Yuzhe Ou, Xujiang Zhao, Jin-Hee Cho, and Feng Chen. 2021. Multidimensional Uncertainty-aware Evidential Neural Networks. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 7815–7822. (Cited on page 56)

Haiwen Huang, Joost van Amersfoort, and Yarin Gal. 2021. Decomposing Representations for Deterministic Uncertainty Estimation. *ArXiv preprint*, abs/2112.00856. (Cited on page 57)

Xinmeng Huang, Shuo Li, Mengxin Yu, Matteo Sesia, Hamed Hassani, Insup Lee, Osbert Bastani, and Edgar Dobriban. 2024. Uncertainty in Language Models: Assessment through Rank-calibration. *ArXiv preprint*, abs/2404.03163. (Cited on page 60)

Yan Huang. 2014. *Pragmatics.* (Cited on page 30)

Yuheng Huang, Jiayang Song, Zhijie Wang, Huaming Chen, and Lei Ma. 2023. Look before You Leap: An Exploratory Study of Uncertainty Measurement for Large Language Models. *ArXiv preprint*, abs/2307.10236. (Cited on page 59)

Eyke Hüllermeier, Sébastien Destercke, and Mohammad Hossein Shaker. 2022. Quantification of Credal Uncertainty in Machine Learning: A Critical Analysis and Empirical Comparison. In *Uncertainty in Artificial Intelligence, Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence, UAI 2022, 1-5 August 2022, Eindhoven, The Netherlands*, volume 180 of *Proceedings of Machine Learning Research*, pages 548–557. (Cited on pages 58 and 168)

Eyke Hüllermeier, Thomas Fober, and Marco Mernberger. 2013. *Inductive Bias*, pages 1018–1018. (Cited on page 8)

Eyke Hüllermeier and Willem Waegeman. 2021. Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods. *Mach. Learn.*, 110(3):457–506. (Cited on page 167)

Anders Humlum and Emilie Vestergaard. 2024. The Adoption of ChatGPT. *University of Chicago, Becker Friedman Institute for Economics Working Paper*, (2024-50). (Cited on page 170)

Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos E. Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, Dennis Ulmer, Florian Schottmann, Khuyagbaatar Batsuren, Kaiser Sun, Koustuv Sinha, Leila Khalatbari, Maria Ryskina, Rita Frieske, Ryan Cotterell, and Zhijing Jin. 2023. A taxonomy and review of generalization research in NLP. *Nature Machine Intelligence*, 5(10):1161–1174. (Cited on pages 99, 113, and 138)

Rasmus Hvingelby, Amalie Brogaard Pauli, Maria Barrett, Christina Rosted, Lasse Malm Lidegaard, and Anders Søgaard. 2020. DaNe: A Named Entity Resource For Danish. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4597–4604. (Cited on pages 114 and 333)

Evgenia Ilia and Wilker Aziz. 2024. Predict the Next Word: < Humans Exhibit Uncertainty in this Task and Language Models _____>. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 234–255. (Cited on pages 60, 64, 162, 167, and 168)

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial Examples Are not Bugs, They Are Features. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 125–136. (Cited on page 77)

Nanna Inie. 2024. What Motivates People to Trust 'AI' Systems? (Cited on page 64)

John P. A. Ioannidis. 2005. Why most Published Research Findings Are False. *PLOS Medicine*, 2(8):null. (Cited on page 82)

Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. (Cited on page 44)

Tovah Irwin, Kyra Wilson, and Alec Marantz. 2023. BERT Shows Garden Path Effects. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3220–3232. (Cited on page 29)

Pavel Izmailov, Wesley Maddox, Polina Kirichenko, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. 2019. Subspace Inference for Bayesian Deep Learning. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pages 1169–1179. (Cited on page 46)

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. 2018. Averaging Weights Leads to Wider Optima and Better Generalization. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 876–885. (Cited on page 46)

Pavel Izmailov, Sharad Vikram, Matthew D. Hoffman, and Andrew Gordon Wilson. 2021. What Are Bayesian Neural Network Posteriors really Like? In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 4629–4640. (Cited on page 38)

Alon Jacovi and Yoav Goldberg. 2021. Aligning Faithful Interpretations with their Social Attribution. *Transactions of the Association for Computational Linguistics*, 9:294–310. (Cited on page 65)

Alon Jacovi, Ana Marasović, Tim Miller, and Yoav Goldberg. 2021. Formalizing Trust in Artificial Intelligence: Prerequisites, Causes and Goals of Human Trust in AI. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 624–635. (Cited on pages 3, 64, 65, and 140)

Siddhartha Jain, Ge Liu, Jonas Mueller, and David Gifford. 2020. Maximizing Overall Diversity for Improved Uncertainty Estimates in Deep Ensembles. In *The Thirty-Fourth AAAI Confer-*

ence on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 4264–4271. (Cited on page 46)

Kalvik Jakkala. 2021. Deep Gaussian Processes: A Survey. *ArXiv preprint*, abs/2106.12135. (Cited on page 47)

Nathalie Japkowicz and Mohak Shah. 2011. *Evaluating Learning Algorithms: A Classification Perspective.* (Cited on page 83)

Alireza Javanmardi, David Stutz, and Eyke Hüllermeier. 2024. Conformalized Credal Set Predictors. *ArXiv preprint*, abs/2402.10723. (Cited on pages 58, 64, and 75)

Frederick Jelinek. 1980. Interpolated Estimation of Markov Source Parameters from Sparse Data. In *Proc. Workshop on Pattern Recognition in Practice, 1980.* (Cited on page 297)

Theis Ingerslev Jensen, Bryan T. Kelly, and Lasse Heje Pedersen. 2021. Is there a Replication Crisis in Finance? Technical report, National Bureau of Economic Research. (Cited on page 71)

Disi Ji, Padhraic Smyth, and Mark Steyvers. 2020. Can I Trust My Fairness Metric? Assessing Fairness with Unlabeled Data and Bayesian Inference. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.* (Cited on page 67)

Haozhe Ji, Pei Ke, Zhipeng Hu, Rongsheng Zhang, and Minlie Huang. 2023a. Tailoring Language Generation Models under Total Variation Distance. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023.* OpenReview.net. (Cited on page 60)

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023b. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12):1–38. (Cited on pages 123 and 171)

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7B. *ArXiv preprint*, abs/2310.06825. (Cited on pages 78 and 90)

Heinrich Jiang, Been Kim, Melody Y. Guan, and Maya R. Gupta. 2018. To Trust or not to Trust a Classifier. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5546–5557. (Cited on page 57)

Nan-Jiang Jiang, Chenhao Tan, and Marie-Catherine de Marneffe. 2023b. Understanding and Predicting Human Label Variation in Natural Language Inference through Explanation. *ArXiv preprint*, abs/2304.12443. (Cited on page 166)

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How Can We Know *When* Language Models Know? On the Calibration of Language Models for Question Answering. *Transactions of the Association for Computational Linguistics (TACL)*, 9:962–977. (Cited on page 156)

Ying Jin and Zhimei Ren. 2024. Confidence on the Focal: Conformal Prediction with Selection-conditional Coverage. *ArXiv preprint*, abs/2403.03868. (Cited on page 166)

Wittawat Jitkrittum, Neha Gupta, Aditya K. Menon, Harikrishna Narasimhan, Ankit Rawat, and Sanjiv Kumar. 2024. When Does Confidence-Based Cascade Deferral Suffice? *Advances in Neural Information Processing Systems*, 36. (Cited on page 68)

Leslie K. John, George Loewenstein, and Drazen Prelec. 2012. Measuring the Prevalence of Questionable Research Practices with Incentives for Truth Telling. *Psychological science*, 23(5):524–532. (Cited on page 71)

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale Similarity Search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547. (Cited on pages 127, 129, 138, and 338)

Taejong Joo, Uijung Chung, and Min-Gwan Seo. 2020. Being Bayesian about Categorical Probability. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4950–4961. (Cited on page 281)

Matt Jordan, Justin Lewis, and Alexandros G. Dimakis. 2019. Provable Certificates for Adversarial Examples: Fitting a Ball in the Union of Polytopes. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14059–14069. (Cited on page 98)

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611. (Cited on pages 90, 141, and 148)

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The State and Fate of Linguistic Diversity and Inclusion in the NLP World. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293. (Cited on page 6)

Daniel Jurafsky and James H. Martin. 2022. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 3rd Ed. Draft.* (Cited on pages 28, 33, and 333)

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language Models (Mostly) Know What They Know. *ArXiv preprint*, abs/2207.05221. (Cited on pages 60 and 62)

Adam Tauman Kalai and Santosh S. Vempala. 2024. Calibrated Language Models Must Hallucinate. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 160–171. ACM. (Cited on page 171)

Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. 2021. AMMUS: A Survey of Transformer-based Pretrained Models in Natural Language Processing. *ArXiv preprint*, abs/2108.05542. (Cited on page 7)

Jenna Kanerva and Filip Ginter. 2022. Out-of-domain Evaluation of Finnish Dependency Parsing. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1114–1124. (Cited on pages 112 and 113)

Archit Karandikar, Nicholas Cain, Dustin Tran, Balaji Lakshminarayanan, Jonathon Shlens, Michael C. Mozer, and Becca Roelofs. 2021. Soft Calibration Objectives for Neural Networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 29768–29779. (Cited on page 37)

Garry Kasparov. 2017. *Deep Thinking: Where Machine Intelligence Ends and Human Creativity Begins*. (Cited on page 170)

Kate Kearns. 2017. *Semantics*. (Cited on pages 26 and 30)

Maurice G. Kendall. 1938. A New Measure of Rank Correlation. *Biometrika*, 30(1/2):81–93. (Cited on page 115)

Christopher Kennedy. 2011. Ambiguity and Vagueness: An Overview. *Semantics: An International Handbook of Natural Language Meaning*, 1:507–535. (Cited on pages 26 and 30)

John Maynard Keynes. 1921. *A Treatise on Probability*. (Cited on page 57)

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest Neighbor Machine Translation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. (Cited on pages 124 and 127)

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through Memorization: Nearest Neighbor Language Models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. (Cited on pages 124 and 127)

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking Benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124. (Cited on page 76)

Hyunsu Kim, Jongmin Yoon, and Juho Lee. 2024a. Fast Ensembling with Diffusion Schrödinger Bridge. *ArXiv preprint*, abs/2404.15814. (Cited on page 46)

Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. 2021. Task-aware Variational Adversarial Active Learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 8166–8175. (Cited on page 69)

Sunnie S. Y. Kim, Q. Vera Liao, Mihaela Vorvoreanu, Stephanie Ballard, and Jennifer Wortman Vaughan. 2024b. I'm Not Sure, But...: Examining the Impact of Large Language Models' Uncertainty Expression on User Reliance and Trust. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency, FAccT 2024, Rio de Janeiro, Brazil, June 3-6, 2024*, pages 822–835. ACM. (Cited on page 66)

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.* (Cited on pages 85, 329, and 332)

Diederik P. Kingma and Max Welling. 2014. Auto-encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings.* (Cited on pages 43 and 166)

John Kirchenbauer, Jacob Oaks, and Eric Heim. 2022. What Is Your Metric Telling You? Evaluating Classifier Calibration under Context-specific Definitions of Reliability. *ArXiv preprint*, abs/2205.11454. (Cited on page 37)

Andreas Kirsch and Yarin Gal. 2022. Unifying Approaches in Active Learning and Active Sampling via Fisher Information and Information-theoretic Quantities. *Transactions on Machine Learning Research (TMLR)*, 2022. (Cited on page 69)

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. 2019. BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7024–7035. (Cited on page 69)

Rebecca Knowles and Philipp Koehn. 2016. Neural Interactive Translation Prediction. In *Conferences of the Association for Machine Translation in the Americas: MT Researchers' Track*, pages 107–120. (Cited on page 130)

Rebecca Knowles, Marina Sanchez-Torron, and Philipp Koehn. 2019. A User Study of Neural Interactive Translation Prediction. *Machine Translation*, 33:135–154. (Cited on page 130)

Olaf Koeneman and Hedde Zeijlstra. 2017. *Introducing Syntax.* (Cited on page 27)

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro

Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran S. Haque, Sara M. Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. 2021. WILDS: A Benchmark of In-the-Wild Distribution Shifts. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5637–5664. (Cited on page 74)

Zahra Kolagar and Alessandra Zarcone. 2024. Aligning Uncertainty: Leveraging LLMs to Analyze Uncertainty Transfer In Text Summarization. In *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024)*, pages 41–61. (Cited on pages 31 and 167)

Andrey Kolmogoroff. 1941. Interpolation und Extrapolation von Stationären Zufälligen Folgen. *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya*, 5(1):3–14. (Cited on page 47)

Benjamin Kompa, Jasper Snoek, and Andrew L Beam. 2021. Empirical Frequentist Coverage of Deep Learning Uncertainty Quantification Procedures. *Entropy*, 23(12):1608. (Cited on pages 39 and 115)

Lingkai Kong, Haoming Jiang, Yuchen Zhuang, Jie Lyu, Tuo Zhao, and Chao Zhang. 2020a. Calibrated Language Model Fine-tuning for In- and Out-of-distribution Data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1326–1340. (Cited on pages 67 and 68)

Lingkai Kong, Jimeng Sun, and Chao Zhang. 2020b. SDE-Net: Equipping Deep Neural Networks with Uncertainty Estimates. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5405–5415. (Cited on pages 56 and 166)

Anna-Kathrin Kopetzki, Bertrand Charpentier, Daniel Zügner, Sandhya Giri, and Stephan Günnemann. 2021. Evaluating Robustness of Predictive Uncertainty Estimation: Are Dirichlet-based Models Reliable? In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5707–5718. (Cited on page 162)

Nikita Kotelevskii, Aleksandr Artemenkov, Kirill Fedyanin, Fedor Noskov, Alexander Fishkov, Artem Shelmanov, Artem Vazhentsev, Aleksandr Petiushko, and Maxim Panov. 2022. Nonparametric Uncertainty Quantification for Single Deterministic Neural Network. *Advances in Neural Information Processing Systems*, 35:36308–36323. (Cited on page 57)

Lea Krause, Wondimagegnhue Tufa, Selene Báez Santamaría, Angel Daza, Urja Khurana, and Piek Vossen. 2023. Confidently Wrong: Exploring the Calibration and Expression of (Un-)Certainty of Large Language Models in a Multilingual Setting. In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*, pages 1–9. (Cited on page 62)

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. Quality at a Glance: An Audit of Web-crawled Multilingual Datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72. (Cited on pages 74 and 77)

Kalimuthu Krishnamoorthy. 2006. *Handbook of Statistical Distributions with Applications.* (Cited on page 16)

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. 2020. Being Bayesian, even just a bit, Fixes Overconfidence in ReLU Networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5436–5446. (Cited on page 45)

David Krueger, Chin-Wei Huang, Riashat Islam, Ryan Turner, Alexandre Lacoste, and Aaron Courville. 2017. Bayesian Hypernetworks. *ArXiv preprint*, abs/1710.04759. (Cited on page 44)

John K. Kruschke. 2013. Bayesian Estimation Supersedes the t test. *Journal of Experimental Psychology: General*, 142(2):573. (Cited on pages 81 and 95)

John K. Kruschke. 2021. Bayesian Analysis Reporting Guidelines. *Nature human behaviour*, 5(10):1282–1291. (Cited on page 81)

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. (Cited on pages 61, 90, 145, 152, and 156)

Thomas S. Kuhn. 1970. *The Structure of Scientific Revolutions*, volume 111. (Cited on page 72)

Meelis Kull, Miquel Perelló-Nieto, Markus Kängsepp, Telmo de Menezes e Silva Filho, Hao Song, and Peter A. Flach. 2019. Beyond Temperature Scaling: Obtaining Well-calibrated Multiclass Probabilities with Dirichlet Calibration. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 12295–12305. (Cited on page 37)

Ananya Kumar, Percy Liang, and Tengyu Ma. 2019. Verified Uncertainty Calibration. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3787–3798. (Cited on page 37)

Aviral Kumar and Sunita Sarawagi. 2019. Calibration of Encoder Decoder Models for Neural Machine Translation. *ArXiv preprint*, abs/1903.00802. (Cited on page 162)

Shankar Kumar and William Byrne. 2002. Minimum Bayes-risk Word Alignments of Bilingual Texts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 140–147. (Cited on page 138)

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk Decoding for Statistical Machine Translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169–176. (Cited on page 138)

Morton Kupperman. 1964. Probabilities of Hypotheses and Information-statistics in Sampling from Exponential-class Populations. *Selected Mathematical Papers*, 29(2):57. (Cited on page 278)

Gleb Kuzmin, Artem Vazhentsev, Artem Shelmanov, Xudong Han, Simon Suster, Maxim Panov, Alexander Panchenko, and Timothy Baldwin. 2023. Uncertainty Estimation for Debiased Models: Does Fairness Hurt Reliability? In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 744–770. (Cited on page 67)

Selim Kuzucu, Jiaee Cheong, Hatice Gunes, and Sinan Kalkan. 2023. Uncertainty-based Fairness Measures. *ArXiv preprint*, abs/2312.11299. (Cited on page 67)

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the Carbon Emissions of Machine Learning. *Workshop on Tackling Climate Change with Machine Learning at NeurIPS 2019.* (Cited on page 326)

Salem Lahlou, Moksh Jain, Hadi Nekoei, Victor Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. 2023. DEUP: Direct Epistemic Uncertainty Prediction. *Transactions on Machine Learning Research*, 2023. (Cited on page 57)

George Lakoff. 1973. Hedges: A Study in Meaning Criteria and the Logic of Fuzzy Concepts. *Journal of Philosophical Logic*, 2(4):458–508. (Cited on page 30)

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6402–6413. (Cited on pages 38, 46, 50, 109, 111, and 116)

Andrew Kyle Lampinen, Stephanie C. Y. Chan, Adam Santoro, and Felix Hill. 2021. Publishing Fast and Slow: A Path Toward Generalizability in Psychology and AI. (Cited on page 82)

Pierre-Simon Laplace. 1774. *Mémoires de Mathématique et de Physique.* (Cited on page 45)

Wassennan Larry. 2004. All of Statistics: A Concise Course in Statistical Inference. (Cited on page 115)

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An Evaluation Dataset for Intent Classification and Out-of-scope Prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316. (Cited on pages 112 and 113)

Olivier Laurent, Adrien Lafage, Enzo Tartaglione, Geoffrey Daniel, Jean-Marc Martinez, Andrei Bursuc, and Gianni Franchi. 2023. Packed Ensembles for Efficient Uncertainty Estimation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. (Cited on page 46)

Benjamin LeBrun, Alessandro Sordoni, and Timothy J. O'Donnell. 2022. Evaluating Distributional Distortion in Neural Language Modeling. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. (Cited on pages 60 and 162)

Noah Lee, Na Min An, and James Thorne. 2023. Can Large Language Models Capture Dissenting Human Voices? In *The 2023 Conference on Empirical Methods in Natural Language Processing*. (Cited on page 167)

Erich Leo Lehmann. 1955. Ordered Families of Distributions. *The Annals of Mathematical Statistics*, pages 399–419. (Cited on page 85)

J. A. Leonard, Mark A. Kramer, and L. H. Ungar. 1992. A Neural Network Architecture that Computes Its own Reliability. *Computers & chemical engineering*, 16(9):819–835. (Cited on page 115)

Thomas Hoskyns Leonard. 2014. A Personal History of Bayesian Statistics. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(2):80–115. (Cited on page 19)

Willem JM Levelt. 1993. *Speaking: From Intention to Articulation*. (Cited on page 30)

A.-M. Leventi-Peetz and T. Östreich. 2022. Deep Learning Reproducibility and Explainable AI (XAI). *ArXiv preprint*, abs/2202.11452. (Cited on page 80)

Esther Levin, Naftali Tishby, and Sara A. Solla. 1990. A Statistical Approach to Learning and Generalization in Layered Neural Networks. *Proceedings of the IEEE*, 78(10):1568–1574. (Cited on page 46)

David D. Lewis and Jason Catlett. 1994. Heterogeneous Uncertainty Sampling for Supervised Learning. In *Machine learning proceedings 1994*, pages 148–156. (Cited on page 68)

David D. Lewis and William A. Gale. 1994. A Sequential Algorithm for Training Text Classifiers. (Cited on page 68)

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and Answer Test-train Overlap in Open-domain Question Answering Datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008. (Cited on page 152)

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A Community Library for Natural Language Processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184. (Cited on page 129)

Bolian Li and Ruqi Zhang. 2023. Entropy-MCMC: Sampling from flat basins with ease. In *NeurIPS 2023 Workshop on Symmetry and Geometry in Neural Representations*. (Cited on page 42)

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2018. Visualizing the Loss Landscape of Neural Nets. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6391–6401. (Cited on page 84)

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. 2017. Hyperband: A Novel Bandit-based Approach to Hyperparameter Optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816. (Cited on pages 78 and 330)

Margaret Y. Li, Alisa Liu, Zhaofeng Wu, and Noah A. Smith. 2024a. A Taxonomy of Ambiguity Types for NLP. *ArXiv preprint*, abs/2403.14072. (Cited on page 32)

Minzhi Li, Taiwei Shi, Caleb Ziems, Min-Yen Kan, Nancy F. Chen, Zhengyuan Liu, and Diyi Yang. 2023a. CoAnnotating: Uncertainty-guided Work Allocation between Human and Large Language Models for Data Annotation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1487–1505. Association for Computational Linguistics. (Cited on page 69)

Moxin Li, Wenjie Wang, Fuli Feng, Fengbin Zhu, Qifan Wang, and Tat-Seng Chua. 2024b. Think Twice before Assure: Confidence Estimation for Large Language Models through Reflection on Multiple Answers. *ArXiv preprint*, abs/2403.09972. (Cited on page 61)

Xiang Lisa Li, Urvashi Khandelwal, and Kelvin Guu. 2024c. Few-Shot Recalibration of Language Models. *ArXiv preprint*, abs/2403.18286. (Cited on page 59)

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023b. Making Language Models Better Reasoners with Step-aware Verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333. (Cited on page 61)

Yingzhen Li and Yarin Gal. 2017. Dropout Inference an Bayesian Neural Networks with Alpha-divergences. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2052–2061. (Cited on page 44)

Kaiqu Liang, Zixu Zhang, and Jaime Fernández Fisac. 2024. Introspective Planning: Guiding Language-enabled Agents to Refine their Own Uncertainty. *ArXiv preprint*, abs/2402.06529. (Cited on page 164)

Q Vera Liao and S Shyam Sundar. 2022. Designing for Responsible Trust in AI Systems: A Communication Perspective. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1257–1268. (Cited on page 65)

Mark Liberman. 2015. Replicability vs. Reproducibility — Or Is It the other Way around? https://languagelog.ldc.upenn.edu/nll/?p=21956. Accessed: 21.02.2022. (Cited on page 72)

Julian Lienen, Caglar Demir, and Eyke Hullermeier. 2023. Conformal Credal Self-supervised Learning. In *Conformal and Probabilistic Prediction with Applications*, pages 214–233. PMLR. (Cited on page 58)

Julian Lienen and Eyke Hüllermeier. 2021a. Credal Self-supervised Learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 14370–14382. (Cited on page 58)

Julian Lienen and Eyke Hüllermeier. 2021b. From Label Smoothing to Label Relaxation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 8583–8591. (Cited on page 37)

Constantine Lignos, Nolan Holley, Chester Palen-Michel, and Jonne Sälevä. 2022. Toward more Meaningful Resources for Lower-resourced Languages. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 523–532. (Cited on page 113)

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81. (Cited on pages 90, 149, and 339)

Jiayu Lin. 2016. On the Dirichlet Distribution. *Master's Report.* (Cited on pages 278, 279, and 281)

Mingfeng Lin, Henry C. Lucas Jr., and Galit Shmueli. 2013. Research Commentary—Too Big to Fail: Large Samples and the p-value Problem. *Information Systems Research*, 24(4):906–917. (Cited on pages 81 and 95)

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022a. Teaching Models to Express their Uncertainty in Words. *Transactions on Machine Learning Research (TMLR).*, 2022. (Cited on pages 59, 62, 144, 146, and 151)

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2022b. Conformal Prediction Intervals with Temporal Dependence. *Transactions on Machine Learnin Research*, 2022. (Cited on page 40)

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with Confidence: Uncertainty Quantification for Black-box Large

Language Models. *ArXiv preprint*, abs/2305.19187. (Cited on pages 63 and 156)

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. Contextualized Sequence Likelihood: Enhanced Confidence Scores For Natural Language Generation. *ArXiv preprint*, abs/2406.01806. (Cited on page 60)

Chen Ling, Xujiang Zhao, Wei Cheng, Yanchi Liu, Yiyou Sun, Xuchao Zhang, Mika Oishi, Takao Osaki, Katsushi Matsuda, Jie Ji, et al. 2024. Uncertainty Decomposition and Quantification for In-context Learning of Large Language Models. *ArXiv preprint*, abs/2402.10189. (Cited on page 61)

Bingyuan Liu, Ismail Ben Ayed, Adrian Galdran, and Jose Dolz. 2022a. The Devil Is in the Margin: Margin-based Label Smoothing for Network Calibration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 80–88. (Cited on page 37)

Chang Liu, Jun Zhu, and Yang Song. 2016. Stochastic Gradient Geodesic MCMC Methods. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3009–3017. (Cited on page 43)

Haitao Liu, Yew-Soon Ong, Xiaomo Jiang, and Xiaofang Wang. 2021. Deep Latent-variable Kernel Learning. *IEEE Transactions on Cybernetics*, 52(10):10276–10289. (Cited on page 47)

Jeremiah Zhe Liu, Shreyas Padhy, Jie Ren, Zi Lin, Yeming Wen, Ghassen Jerfel, Zachary Nado, Jasper Snoek, Dustin Tran, and Balaji Lakshminarayanan. 2023. A Simple Approach to Improve Single-model Deep Uncertainty via Distance-awareness. *Journal of Machine Learning Research (JMLR)*, 24:42:1–42:63. (Cited on pages 68, 111, 300, 332, 334, and 339)

Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. 2024a. Infini-Gram: Scaling Unbounded n-gram Language Models to a Trillion Tokens. *ArXiv preprint*, abs/2401.17377. (Cited on pages 60 and 162)

Linyu Liu, Yu Pan, Xiaocheng Li, and Guanting Chen. 2024b. Uncertainty Estimation and Quantification for LLMs: A Simple Supervised Approach. *ArXiv preprint*, abs/2404.15993. (Cited on pages 57, 164, and 168)

Shiwei Liu, Tianlong Chen, Zahra Atashgahi, Xiaohan Chen, Ghada Sokar, Elena Mocanu, Mykola Pechenizkiy, Zhangyang Wang, and Decebal Constantin Mocanu. 2022b. Deep Ensembling with no Overhead for either Training or Testing: The All-round Blessings of Dynamic Sparsity. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022.* (Cited on page 46)

Yong Liu and Xin Yao. 1999. Ensemble Learning via Negative Correlation. *Neural networks*, 12(10):1399–1404. (Cited on page 46)

Ziyin Liu, Zhikang Wang, Paul Pu Liang, Ruslan Salakhutdinov, Louis-Philippe Morency, and Masahito Ueda. 2019. Deep Gamblers: Learning to Abstain with Portfolio Theory. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 10622–10632. (Cited on page 57)

Joseph J. Locascio. 2017. Results Blind Science Publishing. *Basic and applied social psychology*, 39(5):239–246. (Cited on page 81)

Alexandra Lorson, Chris Cummins, and Hannah Rohde. 2021. Strategic Use of (Un-)Certainty Expressions. *Frontiers in Communication*, 6:635156. (Cited on page 30)

Alexandra Lorson, Hannah Rohde, and Chris Cummins. 2023. Epistemicity and Communicative Strategies. *Discourse Processes*, 60(8):556–593. (Cited on page 30)

Ilya Loshchilov and Frank Hutter. 2018. Fixing Weight Decay Regularization in Adam. (Cited on page 147)

Kadan Lottick, Silvia Susai, Sorelle A. Friedler, and Jonathan P. Wilson. 2019. Energy Usage Reports: Environmental Awareness as Part of Algorithmic Accountability. *Workshop on Tackling Climate Change with Machine Learning at NeurIPS 2019.* (Cited on page 326)

Christos Louizos and Max Welling. 2016. Structured and Efficient Variational Deep Learning with Matrix Gaussian Posteriors. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1708–1716. (Cited on page 44)

Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. 2017. The Expressive Power of Neural Networks: A View from the Width. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6231–6239. (Cited on page 41)

Michal Lukasik, Srinadh Bhojanapalli, Aditya Krishna Menon, and Sanjiv Kumar. 2020. Does Label Smoothing Mitigate Label Noise? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6448–6458. (Cited on page 37)

Yan Luo, Yongkang Wong, Mohan S. Kankanhalli, and Qi Zhao. 2021. Learning to Predict Trustworthiness with Steep Slope Loss. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 21533–21544. (Cited on page 57)

Hengyuan Ma, Yang Qi, Li Zhang, Wenlian Lu, and Jianfeng Feng. 2023. Probabilistic Computation with Emerging Covariance: Towards Efficient Uncertainty Quantification. *ArXiv preprint*, abs/2305.19265. (Cited on page 56)

Xingchen Ma and Matthew B. Blaschko. 2021. Meta-Cal: Well-controlled Post-hoc Calibration by Ranking. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 7235–7245. (Cited on page 37)

Yi-An Ma, Tianqi Chen, and Emily B. Fox. 2015. A Complete Recipe for Stochastic Gradient MCMC. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2917–2925. (Cited on page 43)

David J.C. MacKay. 1992a. A Practical Bayesian Framework for Backpropagation Networks. *Neural computation*, 4(3):448–472. (Cited on page 41)

David J.C. MacKay. 1992b. Bayesian Interpolation. *Neural computation*, 4(3):415–447. (Cited on page 45)

Wesley J. Maddox, Pavel Izmailov, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. 2019. A Simple Baseline for Bayesian Uncertainty in Deep Learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13132–13143. (Cited on page 46)

Tambiama Madiega. 2021. Artificial Intelligence Act. *European Parliament: European Parliamentary Research Service.* (Cited on page 174)

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2023. Post-hoc Interpretability for Neural NLP: A Survey. *ACM Comput. Surv.*, 55(8):155:1–155:42. (Cited on page 4)

Andrey Malinin, Neil Band, Yarin Gal, Mark J. F. Gales, Alexander Ganshin, German Chesnokov, Alexey Noskov, Andrey Ploskonosov, Liudmila Prokhorenkova, Ivan Provilkov, Vatsal Raina, Vyas Raina, Denis Roginskiy, Mariya Shmatova, Panagiotis Tigas, and Boris Yangel. 2021. Shifts: A Dataset of Real Distributional Shift Across Multiple Large-Scale Tasks. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual.* (Cited on page 111)

Andrey Malinin and Mark J. F. Gales. 2018. Predictive Uncertainty Estimation via Prior Networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7047–7058. (Cited on pages 49, 52, 53, 54, 55, 280, and 282)

Andrey Malinin and Mark J. F. Gales. 2019. Reverse KL-Divergence TrainingoOf Prior Networks: Improved Uncertainty and Adversarial Robustness. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14520–14531. (Cited on page 55)

Andrey Malinin and Mark J. F. Gales. 2021. Uncertainty Estimation in Autoregressive Structured Prediction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021.* (Cited on pages 60, 112, and 166)

Andrey Malinin, Bruno Mlodozeniec, and Mark J. F. Gales. 2020. Ensemble Distribution Distillation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.* (Cited on pages 46 and 55)

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 9004–9017. Association for Computational Linguistics. (Cited on pages 61 and 67)

Henry B. Mann and Donald R. Whitney. 1947. On a Test of whether one of two Random Variables Is Stochastically Larger than the other. *The annals of mathematical statistics*, pages 50–60. (Cited on page 87)

Christopher D. Manning. 2015. Last Words: Computational Linguistics and Deep Learning. *Computational Linguistics*, 41(4):701–707. (Cited on page 82)

Lei Mao. 2019. Introduction to Exponential Family. Accessed April 2022. (Cited on page 278)

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. 2021. Scientific Credibility of Machine Translation Research: A Meta-evaluation of 769 Papers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7297–7306. (Cited on pages 71 and 79)

Juan Maroñas, Roberto Paredes, and Daniel Ramos. 2020. Calibration of Deep Probabilistic Models with Decoupled Bayesian Neural Networks. *Neurocomputing*, 407:194–205. (Cited on page 38)

Juan Maroñas, Daniel Ramos, and Roberto Paredes. 2021. On Calibration of Mixup Training for Deep Neural Networks. In *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshops, S+ SSPR 2020, Padua, Italy, January 21–22, 2021, Proceedings*, pages 67–76. Springer. (Cited on page 38)

Antonis Maronikolakis, Philipp Dufter, and Hinrich Schütze. 2021. Wine Is not V I N. On the Compatibility of Tokenizations Across Languages. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2382–2399. (Cited on page 163)

Jorge Martinez-Gil. 2023. A Survey On Legal Question-answering Systems. *Computer Science Review*, 48:100552. (Cited on page 2)

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. 2022. Chunk-based Nearest Neighbor Machine Translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4228–4245. (Cited on pages 124 and 138)

Viera Maslej-Krešňáková, Martin Sarnovskỳ, Peter Butka, and Kristína Machová. 2020. Comparison of Deep Learning Models and Various Text Pre-processing Techniques for the Toxic Comments Classification. *Applied Sciences*, 10(23):8631. (Cited on page 143)

Sergio Matiz and Kenneth E. Barner. 2019. Inductive Conformal Predictor for Convolutional Neural Networks: Applications to Active Learning for Image Classification. *Pattern Recognition*, 90:172–182. (Cited on page 68)

Roger C. Mayer, James H. Davis, and F. David Schoorman. 1995. An Integrative Model of Organizational Trust. *Academy of management review*, 20(3):709–734. (Cited on page 64)

Warren S. McCulloch and Walter Pitts. 1943. A Logical Calculus of the Ideas Immanent in Nervous Activity. *The bulletin of mathematical biophysics*, 5:115–133. (Cited on page 33)

Sharon Bertsch McGrayne. 2011. *The Theory that Would not Die: How Bayes' Rule Cracked the Enigma Code, Hunted down Russian Submarines, & Emerged Triumphant from two Centuries of Controversy.* (Cited on page 19)

Mark F. Medress, Franklin S. Cooper, Jim W. Forgie, C. C. Green, Dennis H. Klatt, Michael H. O'Malley, Edward P. Neuburg, Allen Newell, D. R. Reddy, B. Ritea, et al. 1977. Speech Understanding Systems: Report of a Steering Committee. *Artificial Intelligence*, 9(3):307–316. (Cited on page 135)

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35. (Cited on page 3)

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023. Locally Typical Sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121. (Cited on pages 123 and 138)

Chuizheng Meng, Loc Trinh, Nan Xu, James Enouen, and Yan Liu. 2022a. Interpretability and Fairness Evaluation of Deep Learning

Models on MIMIC-IV Dataset. *Scientific Reports*, 12(1):7166. (Cited on page 3)

Yuxian Meng, Xiaoya Li, Xiayu Zheng, Fei Wu, Xiaofei Sun, Tianwei Zhang, and Jiwei Li. 2022b. Fast Nearest Neighbor Machine Translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 555–565. (Cited on page 124)

Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. 2020. PULSE: Self-supervised Photo Upsampling via Latent Space Exploration of Generative Models. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 2434–2442. (Cited on page 79)

J. L. Mey. 2006. Pragmatics: Overview. *Concise Encyclopedia of Pragmatics*, pages 786–797. (Cited on page 30)

Marco Miani, Frederik Warburg, Pablo Moreno-Muñoz, Nicki Skafte, and Søren Hauberg. 2022. Laplacian Autoencoders for Learning Stochastic Representations. *Advances in Neural Information Processing Systems*, 35:21059–21072. (Cited on page 166)

Sabrina J. Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y Lee, Benoît Sagot, et al. 2021. Between Words and Characters: A Brief History of Open-vocabulary Modeling and Tokenization in NLP. *ArXiv preprint*, abs/2112.10508. (Cited on page 163)

Sabrina J. Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. 2022. Reducing Conversational Agents' Overconfidence through Linguistic Calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872. (Cited on pages 62, 146, and 169)

Jeffrey W. Miller. 2011. (ML 7.7.A2) Expectation of a Dirichlet Random Variable. (Cited on page 278)

Tim Miller. 2019. Explanation in Artificial Intelligence: Insights from the Social Sciences. *Artificial intelligence*, 267:1–38. (Cited on page 65)

Robert William Milne. 1982. Predicting Garden Path Sentences. *Cognitive Science*, 6(4):349–373. (Cited on page 29)

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the Role of Demonstrations: What Makes In-context Learning Work? In *Proceedings of the 2022 Conference on Empirical*

Methods in Natural Language Processing, pages 11048–11064. (Cited on page 91)

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. 2021. Revisiting the Calibration of Modern Neural Networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 15682–15694. (Cited on page 37)

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model Cards for Model Reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229. (Cited on page 79)

John Mitros and Brian Mac Namee. 2019. On the Validity Of Bayesian Neural Networks for Uncertainty Estimation. In *Proceedings for the 27th AIAI Irish Conference on Artificial Intelligence and Cognitive Science, Galway, Ireland, December 5-6, 2019*, volume 2563 of *CEUR Workshop Proceedings*, pages 140–151. (Cited on page 38)

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of What Art? A Call for Multi-prompt LLM Evaluation. *ArXiv preprint*, abs/2401.00595. (Cited on pages 71, 91, 94, and 160)

Shakir Mohamed, Marie-Therese Png, and William Isaac. 2020. Decolonial AI: Decolonial Theory as Sociotechnical Foresight in Artificial Intelligence. *Philosophy & Technology*, 33(4):659–684. (Cited on pages 71 and 171)

Christopher Mohri and Tatsunori Hashimoto. 2024. Language Models with Conformal Factuality Guarantees. *ArXiv preprint*, abs/2402.10978. (Cited on page 164)

Jose G. Moreno-Torres, Troy Raeder, Rocío Alaíz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera. 2012. A Unifying View on Dataset Shift in Classification. *Pattern Recognit.*, 45(1):521–530. (Cited on pages 3, 99, 113, and 138)

Thomas Mortier, Viktor Bengs, Stijn Luca, and Willem Waegeman. 2022. On Calibration of Ensemble-based Credal Predictors. *stat*, 1050:20. (Cited on page 58)

MosaicML NLP Team. 2023. Introducing MPT-7B: A New Standard for Open-Source, Commercially Usable LLMs. Accessed: 2023.05.05. (Cited on page 89)

Azadeh Sadat Mozafari, Hugo Siqueira Gomes, Wilson Leão, and Christian Gagné. 2019. Unsupervised Temperature Scaling: An Unsupervised Post-Processing Calibration Method of Deep Networks. *ArXiv preprint*, abs/1905.00174. (Cited on page 37)

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Jilin Chen, et al. 2023. Controlled Decoding from Language Models. In *Socially Responsible Language Modelling Research*. (Cited on pages 138 and 164)

Jishnu Mukhoti, Puneet K. Dokania, Philip Torr, and Yarin Gal. 2020a. On Batch Normalisation for Approximate Bayesian Inference. In *Third Symposium on Advances in Approximate Bayesian Inference*. (Cited on page 44)

Jishnu Mukhoti, Andreas Kirsch, Joost van Amersfoort, Philip H. S. Torr, and Yarin Gal. 2021. Deterministic Neural Networks with Appropriate Inductive Biases Capture Epistemic and Aleatoric Uncertainty. *ArXiv preprint*, abs/2102.11582. (Cited on pages 57, 111, 112, and 338)

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip H. S. Torr, and Puneet K. Dokania. 2020b. Calibrating Deep Neural Networks Using Focal Loss. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. (Cited on page 37)

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. 2019. When Does Label Smoothing Help? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4696–4705. (Cited on page 37)

Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. 2021. Genre as Weak Supervision for Cross-lingual Dependency Parsing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4786–4802. (Cited on page 113)

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. 2015. Obtaining Well-calibrated Probabilities Using Bayesian Binning. In *Proceedings of the Twenty-Ninth*

AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA, pages 2901–2907. (Cited on pages 36, 114, 146, and 151)

Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. 2021. Understanding the Failure Modes of Out-of-distribution Generalization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021.* (Cited on page 77)

Eric T. Nalisnick, José Miguel Hernández-Lobato, and Padhraic Smyth. 2019a. Dropout as a Structured Shrinkage Prior. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 4712–4722. (Cited on page 44)

Eric T. Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Görür, and Balaji Lakshminarayanan. 2019b. Do Deep Generative Models Know What They Don't Know? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.* (Cited on pages 98 and 113)

Jay Nandy, Wynne Hsu, and Mong-Li Lee. 2020. Towards Maximizing the Representation Gap between In-Domain & Out-Of-Distribution Examples. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.* (Cited on page 55)

Sharan Narang, Hyung Won Chung, Yi Tay, Liam Fedus, Thibault Fevry, Michael Matena, Karishma Malkan, Noah Fiedel, Noam Shazeer, Zhenzhong Lan, Yanqi Zhou, Wei Li, Nan Ding, Jake Marcus, Adam Roberts, and Colin Raffel. 2021. Do Transformer Modifications Transfer Across Implementations and Applications? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5758–5773. (Cited on page 71)

Roberto Navigli. 2009. Word Sense Disambiguation: A Survey. *ACM computing surveys (CSUR)*, 41(2):1–69. (Cited on page 26)

Radford M. Neal. 1995. *Bayesian Learning for Neural Networks*, volume 118. (Cited on pages 41, 42, and 47)

Hermann Ney, Ute Essen, and Reinhard Kneser. 1994. On Structuring Probabilistic Dependences in Stochastic Language Modelling. *Computer Speech & Language*, 8(1):1–38. (Cited on page 297)

Jerzy Neyman. 1937. Outline of a Theory of Statistical Estimation Based on the Classical Theory of Probability. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 236(767):333–380. (Cited on page 15)

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What Can We Learn from Collective Human Opinions on Natural Language Inference Data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9131–9143. (Cited on page 63)

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. 2024. Kernel Language Entropy: Fine-grained Uncertainty Quantification for LLMs from Semantic Similarities. *ArXiv preprint*, abs/2405.20003. (Cited on page 62)

Jeremy Nixon, Michael W. Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. 2019. Measuring Calibration in Deep Learning. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 38–41. (Cited on page 37)

Jongyoun Noh, Hyekang Park, Junghyup Lee, and Bumsub Ham. 2023. RankMixup: Ranking-based Mixup Training for Network Calibration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1358–1368. (Cited on page 38)

Eric W. Noreen. 1989. Computer Intensive Methods for Hypothesis Testing: An Introduction. *Wiley, New York*, 19:21. (Cited on page 87)

Curtis Northcutt, Lu Jiang, and Isaac Chuang. 2021. Confident Learning: Estimating Uncertainty in Dataset Labels. *Journal of Artificial Intelligence Research*, 70:1373–1411. (Cited on page 166)

Sebastian W. Ober, Carl E. Rasmussen, and Mark van der Wilk. 2021. The Promises and Pitfalls of Deep Kernel Learning. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021*, volume 161 of *Proceedings of Machine Learning Research*, pages 1206–1216. (Cited on page 47)

Charles Kay Ogden and Ivor Armstrong Richards. 1923. The Meaning of Meaning: A Study of the Influence of Thought and of the Science of Symbolism. (Cited on pages 29 and 30)

Byung-Doh Oh and William Schuler. 2023. Why Does Surprisal from Larger Transformer-based Language Models Provide a Poorer Fit to Human Reading Times? *Transactions of the Association for Computational Linguistics*, 11:336–350. (Cited on page 29)

Byung-Doh Oh, Shisen Yue, and William Schuler. 2024. Frequency Explains the Inverse Correlation of Large Language Models' Size, Training Data Amount, and Surprisal's Fit to Reading Times. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 1: Long Papers, St. Julian's, Malta, March 17-22, 2024*, pages 2644–2663. Association for Computational Linguistics. (Cited on page 29)

Roberto I. Oliveira, Paulo Orenstein, Thiago Ramos, and João Vitor Romano. 2022. Split Conformal Prediction for Dependent Data. *ArXiv preprint*, abs/2203.15885. (Cited on page 40)

Emre Onal, Klemens Flöge, Emma Caldwell, Arsen Sheverdin, and Vincent Fortuin. Gaussian Stochastic Weight Averaging for Bayesian Low-rank Adaptation of Large Language Models. In *Sixth Symposium on Advances in Approximate Bayesian Inference-Non Archival Track*. (Cited on page 61)

OpenAI. 2022. Introducing ChatGPT. (Cited on pages 63 and 149)

OpenAI. 2023. GPT-4 Technical Report. (Cited on pages 123 and 138)

Manfred Opper and Cédric Archambeau. 2009. The Variational Gaussian Approximation Revisited. *Neural computation*, 21(3):786–792. (Cited on page 43)

Luis A. Ortega, Simón Rodríguez Santana, and Daniel Hernández-Lobato. 2023. Variational Linearized Laplace Approximation for Bayesian Deep Learning. *ArXiv preprint*, abs/2302.12565. (Cited on page 45)

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing Uncertainty in Neural Machine Translation. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3953–3962. (Cited on pages 4 and 60)

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training Language Models to Follow Instructions with Human Feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744. (Cited on page 63)

Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and William Yang Wang. 2023. On the Risk of Misinformation Pollution with Large Language Models. *ArXiv preprint*, abs/2305.13661. (Cited on page 123)

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. 2002. Inductive Confidence Machines for Regression. In *Machine Learning: ECML 2002: 13th European Conference on Machine Learning Helsinki, Finland, August 19–23, 2002 Proceedings 13*, pages 345–356. Springer. (Cited on pages 12, 39, and 124)

Theodore Papamarkou, Maria Skoularidou, Konstantina Palla, Laurence Aitchison, Julyan Arbel, David Dunson, Maurizio Filippone, Vincent Fortuin, Philipp Hennig, Aliaksandr Hubin, et al. 2024. Position Paper: Bayesian Deep Learning in the Age of Large-scale AI. *ArXiv preprint*, abs/2402.00809. (Cited on page 61)

Nicolas Papernot and Patrick McDaniel. 2018. Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning. *ArXiv preprint*, abs/1803.04765. (Cited on page 57)

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. (Cited on pages 135 and 337)

Hyekang Park, Jongyoun Noh, Youngmin Oh, Donghyeon Baek, and Bumsub Ham. 2023. ACLS: Adaptive and Conditional Label Smoothing for Network Calibration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3936–3945. (Cited on page 37)

Seo Yeon Park and Cornelia Caragea. 2022. On the Calibration of Pre-trained Language Models Using Mixup Guided by Area under the Margin and Saliency. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5364–5374. (Cited on page 59)

Younghyun Park, Wonjeong Choi, Soyeong Kim, Dong-Jun Han, and Jaekyun Moon. 2022. Active Learning For Object Detection with Evidential Deep Learning and Hierarchical Uncertainty Aggregation. In *The Eleventh International Conference on Learning Representations*. (Cited on page 69)

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035. (Cited on page 325)

Kanil Patel, William Beluch, Dan Zhang, Michael Pfeiffer, and Bin Yang. 2021. On-manifold Adversarial Data Augmentation Improves Uncertainty Calibration. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 8029–8036. IEEE. (Cited on page 38)

Silviu Paun, Ron Artstein, and Massimo Poesio. 2022. Statistical Methods for Annotation Analysis. *Synthesis Lectures on Human Language Technologies*, 15(1):1–217. (Cited on page 75)

Nick Pawlowski, Andrew Brock, Matthew CH Lee, Martin Rajchl, and Ben Glocker. 2017. Implicit Weight Uncertainty in Neural Networks. *ArXiv preprint*, abs/1711.01297. (Cited on page 44)

Tim Pearce, Felix Leibfried, and Alexandra Brintrup. 2020. Uncertainty in Neural Networks: Approximately Bayesian Ensembling. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 234–244. (Cited on pages 46 and 109)

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-Learn: Machine Learning in Python. *the Journal of machine Learning research*, 12:2825–2830. (Cited on pages 108, 325, and 329)

Roger D. Peng. 2011. Reproducible Research in Computational Science. *Science*, 334(6060):1226–1227. (Cited on page 72)

Álvaro Peris, Miguel Domingo, and Francisco Casacuberta. 2017. Interactive Neural Machine Translation. *Computer Speech & Language*, 45:201–220. (Cited on page 130)

Dana Pessach and Erez Shmueli. 2023. Algorithmic Fairness. In *Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook*, pages 867–886. (Cited on page 67)

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021. UNKs Everywhere: Adapting Multilingual Language Models to New Scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203. (Cited on page 163)

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaïd Harchaoui. 2021. MAUVE: Measuring the Gap between Neural Text and Human Text Using Divergence Frontiers. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 4816–4828. (Cited on pages 135 and 338)

Silviu Pitis, Michael R Zhang, Andrew Wang, and Jimmy Ba. 2023. Boosted Prompt EnsemblesfFor Large Language Models. *ArXiv preprint*, abs/2304.05970. (Cited on page 61)

Barbara Plank. 2016. What to do About Non-standard (or Non-canonical) Language in NLP. In *Proceedings of the 13th Conference on Natural Language Processing, KONVENS 2016, Bochum, Germany, September 19-21, 2016*, volume 16 of *Bochumer Linguistische Arbeitsberichte*. (Cited on page 6)

Barbara Plank. 2022. The "Problem" of Human Label Variation: On Ground Truth in Data, Modeling and Evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682. (Cited on pages 63, 75, 159, and 165)

Barbara Plank, Kristian Nørgaard Jensen, and Rob van der Goot. 2020. DaN+: Danish Nested Named Entities and Lexical Normalization. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6649–6662. (Cited on pages 112 and 113)

John Platt et al. 1999. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods.

Advances in large margin classifiers, 10(3):61–74. (Cited on pages 37 and 151)

Benjamin Plaut, Khanh Nguyen, and Tu Trinh. 2024. Softmax Probabilities (Mostly) Predict Large Language Model Correctness on Multiple-choice Q&A. *ArXiv preprint*, abs/2402.13213. (Cited on page 59)

Maja Popović. 2017. ChrF++: Words Helping Character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618. (Cited on page 135)

Karl Popper. 1934. *Logik der Forschung*. (Cited on page 72)

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191. (Cited on page 78)

Janis Postels, Mattia Segù, Tao Sun, Luca Daniel Sieber, Luc Van Gool, Fisher Yu, and Federico Tombari. 2022. On the Practicality of Deterministic Epistemic Uncertainty. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 17870–17909. (Cited on page 57)

Vinodkumar Prabhakaran, Aida Mostafazadeh Davani, and Mark Diaz. 2021. On Releasing Annotator-level Labels and Information in Datasets. In *Proceedings of the Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 133–138. (Cited on page 75)

Ellen F. Prince, Joel Frader, Charles Bosk, et al. 1982. On Hedging in Physician-Physician Discourse. *Linguistics and the Professions*, 8(1):83–97. (Cited on page 30)

James Pustejovsky. 1991. The Generative Lexicon. *Computational Linguistics*, 17(4):409–441. (Cited on pages 26 and 27)

James Pustejovsky. 2017. The Semantics of Lexical Underspecification. *Folia linguistica*, 51(s1000):1–25. (Cited on pages 26 and 27)

James Pustejovsky and Amber Stubbs. 2012. *Natural Language Annotation for Machine Learning: A Guide to Corpus-building for Applications*. (Cited on page 75)

Sampo Pyysalo, Jenna Kanerva, Anna Missilä, Veronika Laippala, and Filip Ginter. 2015. Universal Dependencies For Finnish. In

Proceedings of the 20th Nordic Conference of Computational Linguistics (NoDaLiDa 2015), pages 163–172. (Cited on pages 112 and 113)

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. 2022. Cold Decoding: Energy-based Constrained Text Generation with Langevin Dynamics. *Advances in Neural Information Processing Systems*, 35:9538–9551. (Cited on pages 138 and 164)

Xin Qiu and Risto Miikkulainen. 2022. Detecting Misclassification Errors in Neural Networks with a Gaussian Process Model. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 8017–8027. (Cited on page 57)

Eric Qu, Xufang Luo, and Dongsheng Li. 2022. Data Continuity Matters: Improving Sequence Modeling with Lipschitz Regularizer. In *The Eleventh International Conference on Learning Representations*. (Cited on page 6)

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S. Jaakkola, and Regina Barzilay. 2023. Conformal language modeling. In *The Twelfth International Conference on Learning Representations*. (Cited on pages 163, 164, and 165)

Maurice H. Quenouille. 1949. Approximate Tests of Correlation In Time-series. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 45, pages 483–484. Cambridge University Press. (Cited on page 17)

Christopher B. Quirk. 2004. Training a Sentence-level Machine Translation Confidence Measure. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*. (Cited on page 143)

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language Models Are Unsupervised Multitask Learners. *OpenAI blog*, 1(8):9. (Cited on pages 63, 92, 135, and 173)

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-text Transformer. *Journal of Machine Learning Resesarch (JMLR)*, 21:140:1–140:67. (Cited on page 62)

Rahul Rahaman and Alexandre H. Thiéry. 2021. Uncertainty Quantification and Deep Ensembles. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 20063–20075. (Cited on page 38)

Abhinav Ramesh Kashyap, Devamanyu Hazarika, Min-Yen Kan, and Roger Zimmermann. 2021. Domain Divergences: A Survey and Empirical Analysis. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1830–1849. (Cited on page 74)

Sebastian Raschka. 2018. Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. *ArXiv preprint*, abs/1811.12808. (Cited on page 83)

Shauli Ravfogel, Yoav Goldberg, and Jacob Goldberger. 2023. Conformal Nucleus Sampling. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 27–34. Association for Computational Linguistics. (Cited on pages 59, 127, 129, 130, 133, 135, 136, 163, and 165)

Amy Rechkemmer and Ming Yin. 2022. When Confidence Meets Accuracy: Exploring the Effects of Multiple Performance Indicators on Trust in Machine Learning Models. In *Proceedings of the 2022 chi conference on human factors in computing systems*, pages 1–14. (Cited on page 65)

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQa: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266. (Cited on pages 141 and 148)

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 Submission for the Metrics Shared-task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585. (Cited on page 135)

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A Neural Framework for MT Evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702. (Cited on pages 135 and 337)

Nils Reimers and Iryna Gurevych. 2018. Why Comparing Single Performance Scores Does not Allow to Draw Conclusions about Machine Learning Approaches. *ArXiv preprint*, abs/1803.09578. (Cited on page 95)

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings Using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992. (Cited on pages 147 and 149)

Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. 2023. Robots that ask for help: Uncertainty alignment for large language model planners. In *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, pages 661–682. PMLR. (Cited on pages 59 and 164)

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J. Liu. 2022. Out-of-distribution Detection and Selective Generation for Conditional Language Models. In *The Eleventh International Conference on Learning Representations*. (Cited on pages 162 and 164)

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. 2021. A Survey of Deep Active Learning. *ACM computing surveys (CSUR)*, 54(9):1–40. (Cited on page 68)

Valerie F. Reyna and Charles J. Brainerd. 2008. Numeracy, Ratio Bias, and Denominator Neglect in Judgments of Risk and Probability. *Learning and individual differences*, 18(1):89–107. (Cited on page 66)

Danilo Jimenez Rezende and Shakir Mohamed. 2015. Variational Inference with Normalizing Flows. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1530–1538. (Cited on page 56)

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China,* *21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1278–1286. (Cited on pages 43 and 166)

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912. (Cited on page 76)

Stefan Riezler and Michael Hagmann. 2021. Validity, Reliability, and Significance. (Cited on page 83)

Hippolyt Ritter, Aleksandar Botev, and David Barber. 2018a. A Scalable Laplace Approximation for Neural Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. (Cited on page 45)

Hippolyt Ritter, Aleksandar Botev, and David Barber. 2018b. Online Structured Laplace Approximations for Overcoming Catastrophic Forgetting. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 3742–3752. (Cited on page 45)

Christian P. Robert, George Casella, and George Casella. 1999. *Monte Carlo Statistical Methods*, volume 2. (Cited on page 42)

Rebecca Roelofs, Nicholas Cain, Jonathon Shlens, and Michael C. Mozer. 2022. Mitigating Bias in Calibration Error Estimation. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, volume 151 of *Proceedings of Machine Learning Research*, pages 4036–4054. (Cited on page 37)

Anna Rogers and Isabelle Augenstein. 2021. How to Review for ACL Rolling Review? https://aclrollingreview.org/reviewertutorial. Accessed: 2022.02.21. (Cited on page 82)

Alex Rogozhnikov. 2022. Einops: Clear and Reliable Tensor Manipulations with Einstein-like Notation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. (Cited on page 325)

Yaniv Romano, Matteo Sesia, and Emmanuel J. Candès. 2020. Classification with valid and Adaptive Coverage. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. (Cited on page 128)

Frank Rosenblatt. 1958. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological review*, 65(6):386. (Cited on page 33)

Frank Rosenblatt et al. 1962. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, volume 55. (Cited on page 33)

Vlada Rozova, Katrina Witt, Jo Robinson, Yan Li, and Karin Verspoor. 2022. Detection of Self-harm And Suicidal Ideation in Emergency Department Triage Notes. *Journal of the American Medical Informatics Association*, 29(3):472–480. (Cited on page 2)

Victoria L. Rubin. 2006. Identifying Certainty in Texts. *Unpublished Doctoral Thesis, Syracuse University, Syracuse, NY.* (Cited on page 30)

Sebastian Ruder. 2020. Why You Should Do NLP Beyond English. http://ruder.io/nlp-beyond-english. (Cited on page 6)

David Ruhe, Giovanni Cina, Michele Tonutti, Daan de Bruin, and Paul Elbers. 2019. Bayesian Modelling in Practice: Using Uncertainty to Improve Trustworthiness in Medical Applications. *ArXiv preprint*, abs/1906.08619. (Cited on page 111)

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How Good Is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135. (Cited on pages 75 and 163)

Erfan Sadeqi Azer, Daniel Khashabi, Ashish Sabharwal, and Dan Roth. 2020. Not all Claims Are Created Equal: Choosing the Right Statistical Approach to Assess Hypotheses. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5715–5725. (Cited on pages 80, 83, and 95)

Yusuf Sale, Viktor Bengs, Michele Caprio, and Eyke Hüllermeier. 2023a. Second-order Uncertainty Quantification: A Distance-based Approach. *ArXiv preprint*, abs/2312.00995. (Cited on page 168)

Yusuf Sale, Michele Caprio, and Eyke Hüllermeier. 2023b. Is the Volume of a Credal Set a Good Measure for Epistemic Uncertainty? In *Uncertainty in Artificial Intelligence*, pages 1795–1804. PMLR. (Cited on page 58)

Yusuf Sale, Paul Hofman, Lisa Wimmer, Eyke Hüllermeier, and Thomas Nagler. 2024. Second-order Uncertainty Quantification: Variance-based Measures. *ArXiv preprint*, abs/2401.00276. (Cited on page 168)

Maarten Sap, Swabha Swayamdipta, Laura Vianna, Xuhui Zhou, Yejin Choi, and Noah A. Smith. 2022. Annotators with Attitudes: How Annotator Beliefs and Identities Bias Toxic Language Detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5884–5906. (Cited on page 74)

Leonard J. Savage. 1971. Elicitation of Personal Probabilities and Expectations. *Journal of the American Statistical Association*, 66(336):783–801. (Cited on page 37)

Walter J. Savitch, Emmon Bach, W. E. Marsh, and Gila Safran-Naveh. 2012. *The Formal Complexity of Natural Language*, volume 33. (Cited on page 28)

Andrew M. Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D. Tracey, and David D. Cox. 2018. On the Information Bottleneck Theory of Deep Learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. (Cited on page 161)

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176B-Parameter Open-access Multilingual Language Model. *ArXiv preprint*, abs/2211.05100. (Cited on page 123)

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. Active Hidden Markov Models for Information Extraction. In *International symposium on intelligent data analysis*, pages 309–318. Springer. (Cited on page 68)

David Schlangen. 2021. Targeting the Benchmark: On Methodology in Current Natural Language Processing Research. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 670–674. (Cited on pages 73 and 77)

Victor Schmidt, Kamal Goyal, Aditya Joshi, Boris Feld, Liam Conell, Nikolas Laskaris, Doug Blank, Jonathan Wilson, Sorelle

Friedler, and Sasha Luccioni. 2021. Codecarbon: Estimate and Track Carbon Emissions from Machine Learning Computing. (Cited on page 326)

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. 2022. Confident Adaptive Language Modeling. *Advances in Neural Information Processing Systems*, 35:17456–17472. (Cited on pages 4, 59, 68, 164, and 165)

Tal Schuster, Adam Fisch, Tommi Jaakkola, and Regina Barzilay. 2021. Consistent Accelerated Inference via Confident Adaptive Transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4962–4979. (Cited on page 4)

Carson T. Schütze. 1995. PP Attachmenta And Argumenthood. *MIT Working Papers in Linguistics*, 26(95):151. (Cited on page 28)

Hinrich Schütze. 1997. Ambiguity Resolution in Language Learning. *CSLI Lecture Notes*, 71. (Cited on page 26)

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI. *Commun. ACM*, 63(12):54–63. (Cited on page 71)

Pola Schwöbel, Martin Jørgensen, Sebastian W. Ober, and Mark van der Wilk. 2022. Last-layer Marginal Likelihood for Invariance Learning. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, volume 151 of *Proceedings of Machine Learning Research*, pages 3542–3555. (Cited on page 47)

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to Start Worrying about Prompt Formatting. In *The Twelfth International Conference on Learning Representations.* (Cited on pages 91 and 160)

Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D. Manning. 2019. Do Massively Pretrained Language Models Make Better Storytellers? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 843–861. (Cited on page 123)

Florian Seligmann, Philipp Becker, Michael Volpp, and Gerhard Neumann. 2024. Beyond Deep Ensembles: A Large-scale Evaluation of Bayesian Deep Learning under Distribution Shift. *Advances in Neural Information Processing Systems*, 36. (Cited on page 38)

Murat Sensoy, Lance M. Kaplan, Federico Cerutti, and Maryam Saleki. 2020. Uncertainty-aware Deep Classifiers Using Generative Models. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 5620–5627. (Cited on pages 52 and 56)

Murat Sensoy, Lance M. Kaplan, and Melih Kandemir. 2018. Evidential Deep Learning to Quantify Classification Uncertainty. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 3183–3193. (Cited on pages 49, 50, 52, 54, 55, 56, 112, 281, and 329)

Burr Settles. 2009. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison. (Cited on page 68)

Jaime Sevilla, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos. 2022. Compute Trends Across Three Eras of Machine Learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE. (Cited on pages 7 and 82)

Uri Shaham and Omer Levy. 2022. What Do You Get when You Cross Beam Search with Nucleus Sampling? In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 38–45. (Cited on page 136)

Yilin Shen, Wenhu Chen, and Hongxia Jin. 2020. Modeling Token-level Uncertainty to Learn Unknown Concepts in SLU via Calibrated Dirichlet Prior RNN. *ArXiv preprint*, abs/2010.08101. (Cited on pages 56 and 59)

Hidetoshi Shimodaira. 2000. Improving Predictive Inference under Covariate Shift by Weighting the Log-likelihood Function. *Journal of statistical planning and inference*, 90(2):227–244. (Cited on pages 3 and 99)

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. Model Dementia: Generated

Data Makes Models Forget. *arXiv e-prints*, pages arXiv–2305. (Cited on page 172)

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan L. Boyd-Graber, and Lijuan Wang. 2023. Prompting GPT-3 to be reliable. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. (Cited on page 59)

Chenglei Si, Chen Zhao, Sewon Min, and Jordan Boyd-Graber. 2022. Re-examining Calibration: The Case of Question Answering. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2814–2829. (Cited on page 59)

Anthony Sicilia, Hyunwoo Kim, Khyathi Raghavi Chandu, Malihe Alikhani, and Jack Hessel. 2024. Deal, or no Deal (or Who Knows)? Forecasting Uncertainty in Conversations Using Large Language Models. *ArXiv preprint*, abs/2402.03284. (Cited on pages 60 and 166)

Herbert A. Simon. 1995. Artificial Intelligence: An Empirical Science. *Artificial Intelligence*, 77(1):95–127. (Cited on page 72)

Aniket Kumar Singh, Bishal Lamichhane, Suman Devkota, Uttam Dhakal, and Chandra Dhakal. 2024a. Do Large Language Models Show Human-like Biases? Exploring Confidence—competence Gap in AI. *Information*, 15(2):92. (Cited on page 62)

Ashudeep Singh, David Kempe, and Thorsten Joachims. 2021. Fairness in Ranking under Uncertainty. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 11896–11908. (Cited on page 67)

Jasdeep Singh, Bryan McCann, Richard Socher, and Caiming Xiong. 2019. BERT is not an Interlingua and the Bias of Tokenization. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 47–55. (Cited on page 163)

Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura OMahony, et al. 2024b. Aya Dataset: An Open-access Collection for Multilingual Instruction Tuning. *ArXiv preprint*, abs/2402.06619. (Cited on page 172)

Siddharth Singi, Zhanpeng He, Alvin Pan, Sandip Patel, Gunnar A. Sigurdsson, Robinson Piramuthu, Shuran Song, and Matei Ciocarlie. 2023. Decision Making for Human-in-the-loop Robotic Agents via Uncertainty-aware Reinforcement Learning. *ArXiv preprint*, abs/2303.06710. (Cited on page 69)

Miroslav Sirota and Marie Juanchich. 2015. A Direct and Comprehensive Test of two Postulates of Politeness Theory Applied to Uncertainty Communication. *Judgment and Decision Making*, 10(3):232–240. (Cited on page 30)

Sarath Sivaprasad and Mario Fritz. 2023. Going beyond Familiar Features for Deep Anomaly Detection. *ArXiv preprint*, abs/2310.00797. (Cited on page 161)

Christian Sivertsen, Guido Salimbeni, Anders Sundnes Løvlie, Steven David Benford, and Jichen Zhu. 2024. Machine Learning Processes as Sources of Ambiguity: Insights from AI Art. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24. (Cited on page 170)

John Skilling and S. Sibisi. 1990. Fundamentals of Maxent in Data-analysis. In *Institute of Physics Conference Series*, 107, pages 1–21. IOP PUBLISHING LTD TEMPLE CIRCUS, TEMPLE WAY, BRISTOL BS1 6BE, ENGLAND. (Cited on page 19)

Freddie Bickford Smith, Andreas Kirsch, Sebastian Farquhar, Yarin Gal, Adam Foster, and Tom Rainforth. 2023. Prediction-oriented Bayesian Active Learning. In *International Conference on Artificial Intelligence and Statistics*, pages 7331–7348. PMLR. (Cited on page 69)

Lewis Smith and Yarin Gal. 2018. Understanding Measures of Uncertainty for Adversarial Example Detection. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 560–569. (Cited on pages 48, 101, 109, and 110)

Michael Smithson. 2003. *Confidence Intervals*. 140. (Cited on page 16)

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 2960–2968. (Cited on pages 78, 148, and 332)

Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D. Sculley, Joshua V. Dillon, Jie Ren, and Zachary Nado. 2019. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty under Dataset Shift. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13969–13980. (Cited on pages 68, 98, 99, 110, 111, 116, and 133)

Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Md. Mostofa Ali Patwary, Prabhat, and Ryan P. Adams. 2015. Scalable Bayesian Optimization Using Deep Neural Networks. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2171–2180. (Cited on page 45)

Anders Søgaard, Sebastian Ebert, Jasmijn Bastings, and Katja Filippova. 2021. We Need to TalkaAbout Random Splits. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1823–1832. (Cited on pages 71 and 76)

Lucia Specia. 2021. Disagreement in Human Evaluation: Blame the Task, not the Annotators. NoDaLiDa Keynote. (Cited on page 75)

David Spiegelhalter. 2017. Risk and Uncertainty Communication. *Annual Review of Statistics and Its Application*, 4:31–60. (Cited on page 66)

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958. (Cited on page 44)

Maximilian Stadler, Bertrand Charpentier, Simon Geisler, Daniel Zügner, and Stephan Günnemann. 2021. Graph Posterior Network: Bayesian Predictive Uncertainty for Node Classification. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 18033–18048. (Cited on pages 56 and 68)

Kamile Stankeviciute, Ahmed M. Alaa, and Mihaela van der Schaar. 2021. Conformal Time-series Forecasting. In *Advances in Neural Information Processing Systems 34: Annual Conference on*

Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pages 6216–6228. (Cited on page 40)

Elias Stengel-Eskin, Peter Hase, and Mohit Bansal. 2024. LACIE: Listener-aware Finetuning for Confidence Calibration in Large Language Models. *ArXiv preprint*, abs/2405.21028. (Cited on page 169)

Jonathon Stewart, Juan Lu, Adrian Goudie, Glenn Arendts, Shiv A. Meka, Sam Freeman, Katie Walker, Peter Sprivulis, Frank Sanfilippo, Mohammed Bennamoun, et al. 2022. Applications of Natural Language Processing at Emergency Department Triage: A Systematic Review. *medRxiv*, pages 2022–12. (Cited on page 2)

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to Summarize with Human Feedback. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. (Cited on pages 63 and 339)

Andreas Stolcke. 2002. SRILM - An Extensible Language Modeling Toolkit. In *Seventh international conference on spoken language processing*. (Cited on page 297)

Rebecca S. Stone, Nishant Ravikumar, Andrew J. Bulpitt, and David C. Hogg. 2022. Epistemic Uncertainty-weighted Loss for Visual Bias Mitigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2022, New Orleans, LA, USA, June 19-20, 2022*, pages 2897–2904. (Cited on page 67)

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650. (Cited on page 71)

Patrick Sturt, Martin J Pickering, and Matthew W Crocker. 1999. Structural Change and Reanalysis Difficulty in Language Comprehension. *Journal of Memory and Language*, 40(1):136–150. (Cited on page 28)

Jiayuan Su, Jing Luo, Hongwei Wang, and Lu Cheng. 2024. API Is Enough: Conformal Prediction for Large Language Models without Logit-access. *ArXiv preprint*, abs/2403.01216. (Cited on pages 63, 164, and 165)

Hao Sun, Boris van Breugel, Jonathan Crabbé, Nabeel Seedat, and Mihaela van der Schaar. 2024. What Is Flagged in Uncertainty Quantification? Latent Density Models for Uncertainty Categorization. *Advances in Neural Information Processing Systems*, 36. (Cited on page 57)

Richard Sutton. 2019. The Bitter Lesson. *Incomplete Ideas (blog)*, 13:12. (Cited on page 171)

Benjamin Swets, Timothy Desmet, Charles Clifton, and Fernanda Ferreira. 2008. Underspecification of Syntactic Ambiguities: Evidence from Self-paced Reading. *Memory & Cognition*, 36:201–216. (Cited on page 29)

Zoltán Gendler Szabó. 2004. Compositionality. (Cited on page 27)

György Szarvas, Veronika Vincze, Richárd Farkas, György Móra, and Iryna Gurevych. 2012. Cross-genre and Cross-domain Detection of Semantic Uncertainty. *Computational Linguistics*, 38(2):335–367. (Cited on pages 31 and 167)

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. (Cited on page 37)

Natasa Tagasovska and David Lopez-Paz. 2019. Single-model Uncertainties for Deep Learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6414–6425. (Cited on page 112)

Abdul Karim Taha. 1983. Types of Syntactic Ambiguity in English. (Cited on page 28)

Anique Tahir, Lu Cheng, and Huan Liu. 2023. Fairness through Aleatoric Uncertainty. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 2372–2381. (Cited on page 67)

Linwei Tao, Minjing Dong, and Chang Xu. 2023. Dual Focal Loss for Calibration. In *International Conference on Machine Learning*, pages 33833–33849. PMLR. (Cited on page 37)

Yi Tay, Mostafa Dehghani, Samira Abnar, Hyung Won Chung, William Fedus, Jinfeng Rao, Sharan Narang, Vinh Q. Tran, Dani Yogatama, and Donald Metzler. 2023. Scaling Laws vs. Model Architectures: How does Inductive Bias Influence Scaling? In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 12342–12364. Association for Computational Linguistics. (Cited on page 6)

Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. 2016. Branchynet: Fast Inference via Early Exiting from Deep Neural Networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 2464–2469. IEEE. (Cited on page 68)

Mattias Teye, Hossein Azizpour, and Kevin Smith. 2018. Bayesian Uncertainty Estimation for Batch Normalized Deep Networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4914–4923. (Cited on pages 44 and 45)

The Decoder. 2023. GPT-4 Architecture, Datasets, Costs and more Leaked. [Online; accessed 02-May-2024]. (Cited on pages 139 and 174)

The United States District Court for the S.D.N.Y. 2013. Roberto Mata v. Aviance Inc. (Cited on page 170)

Sunil Thulasidasan, Gopinath Chennupati, Jeff A. Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. 2019. On Mixup Training: Improved Calibration and Predictive Uncertainty for Deep Neural Networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13888–13899. (Cited on page 38)

Arthur Thuy and Dries F. Benoit. 2023. Explainability through Uncertainty: Trustworthy Decision-making with Neural Networks. *European Journal of Operational Research*. (Cited on page 67)

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. 2023. Just Ask for Calibration: Strategies for Eliciting Calibrated Confidence Scores from Language Models Fine-tuned with Human Feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5433–5442. (Cited on pages 62, 144, 146, 151, 169, and 172)

Robert J. Tibshirani and Bradley Efron. 1993. An Introduction to the Bootstrap. *Monographs on statistics and applied probability*, 57(1):1–436. (Cited on pages 14 and 17)

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-RMSprop: Divide the Gradient by a Running Average of Its Recent Magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31. (Cited on page 85)

William Timkey and Marten van Schijndel. 2021. All Bark and no Bite: Rogue Dimensions in Transformer Language Models Obscure Representational Quality. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4527–4546. (Cited on page 147)

Tishby and Solla. 1989. Consistent Inference of Probabilities in Layered Networks: Predictions and Generalizations. In *International 1989 joint conference on neural networks*, pages 403–409. IEEE. (Cited on page 41)

Naftali Tishby, Fernando C. Pereira, and William Bialek. 2000. The Information Bottleneck Method. *arXiv preprint physics/0004057*. (Cited on page 161)

Naftali Tishby and Noga Zaslavsky. 2015. Deep Learning and the Information Bottleneck Principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE. (Cited on page 161)

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and Efficient Foundation Language Models. *ArXiv preprint*, abs/2302.13971. (Cited on pages 78, 123, and 149)

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open Foundation and Fine-tuned Chat Models. *ArXiv preprint*, abs/2307.09288. (Cited on page 78)

Alexander Treiss, Jannis Walk, and Niklas Kühl. 2021. An Uncertainty-based Human-in-the-loop System for Industrial Tool Wear Analysis. In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part V*, pages 85–100. Springer. (Cited on page 69)

Tina Tseng, Amanda Stent, and Domenic Maida. 2020. Best Practices for Managing Data Annotation Projects. *ArXiv preprint*, abs/2009.11654. (Cited on page 75)

Theodoros Tsiligkaridis. 2019. Information Robust Dirichlet Networks for Predictive Uncertainty Estimation. *ArXiv preprint*, abs/1910.04819. (Cited on page 55)

Stéphane Tufféry. 2011. *Data Mining and Statistics for Decision Making*. (Cited on page 79)

David Tuggy. 1993. Ambiguity, Polysemy, and Vagueness. (Cited on page 26)

John Tukey. 1958. Bias and Confidence in not Quite Large Samples. *Ann. Math. Statist.*, 29:614. (Cited on page 17)

A.M. Turing. 1950. Computing Machinery and Intelligence. (Cited on page 5)

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. (Cited on pages 144 and 154)

Dennis Ulmer. 2019. Recoding Latent Sentence Representations–Dynamic Gradient-based Activation Modification in RNNs. *ArXiv preprint*, abs/2101.00674. (Cited on page 173)

Dennis Ulmer, Elisa Bassignana, Max Müller-Eberstein, Daniel Varab, Mike Zhang, Rob van der Goot, Christian Hardmeier, and Barbara Plank. 2022a. Experimental Standards for Deep Learning in Natural Language Processing Research. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2673–2692. (Cited on pages 11, 72, and 325)

Dennis Ulmer and Giovanni Cinà. 2021. Know Your Limits: Uncertainty Estimation with ReLU Classifiers Fails at Reliable OOD Detection. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021*, volume 161 of *Proceedings of Machine Learning Research*, pages 1766–1776. (Cited on pages 11, 68, and 98)

Dennis Ulmer, Jes Frellsen, and Christian Hardmeier. 2022b. Exploring Predictive Uncertainty and Calibration in NLP: A Study on the Impact of Method & Data Scarcity. In *Findings of the*

Association for Computational Linguistics: EMNLP 2022, pages 2707–2735. (Cited on pages 11, 59, 68, and 110)

Dennis Ulmer, Martin Gubri, Hwaran Lee, Sangdoo Yun, and Seong Joon Oh. 2024a. Calibrating Large Language Models Using Their Generations only. *arXiv preprint 2403.05973*. (Cited on pages 12, 59, 60, 62, 63, 140, and 172)

Dennis Ulmer, Christian Hardmeier, and Jes Frellsen. 2022c. Deep-significance: Easy and Meaningful Signifcance Testing in the Age of Neural Networks. In *ML Evaluation Standards Workshop at the Tenth International Conference on Learning Representations*. (Cited on pages 11, 80, 83, 136, 137, and 151)

Dennis Ulmer, Christian Hardmeier, and Jes Frellsen. 2023. Prior and Posterior Networks: A Survey on Evidential Deep Learning Methods for Uncertainty Estimation. *Transactions on Machine Learning Research*. (Cited on pages 11, 48, and 55)

Dennis Ulmer, Elman Mansimov, Kaixiang Lin, Justin Sun, Xibin Gao, and Yi Zhang. 2024b. Bootstrapping LLM-based Task-oriented Dialogue Agents via Self-talk. *ArXiv preprint*, abs/2401.05033. (Cited on page 172)

Dennis Ulmer, Lotta Meijerink, and Giovanni Cinà. 2020. Trust Issues: Uncertainty Estimation Does not Enable Reliable OOD Detection on Medical Tabular Data. In *Machine Learning for Health*, pages 341–354. PMLR. (Cited on pages 68, 98, 99, 110, and 111)

Dennis Ulmer, Chrysoula Zerva, and André F. T. Martins. 2024c. Non-exchangeable conformal language generation with nearest neighbors. In *Findings of the Association for Computational Linguistics: EACL 2024, St. Julian's, Malta, March 17-22, 2024*, pages 1909–1929. Association for Computational Linguistics. (Cited on pages 12, 59, 124, and 165)

Alexandra N. Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. Learning from Disagreement: A Survey. *Journal of Artificial Intelligence Research*, 72:1385–1470. (Cited on pages 63 and 75)

Anshuk Uppal, Kristoffer Stensbo-Smidt, Wouter Boomsma, and Jes Frellsen. 2024. Implicit Variational Inference for High-dimensional Posteriors. *Advances in Neural Information Processing Systems*, 36. (Cited on page 68)

Joost van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. 2021. On Feature Collapse and Deep Kernel Learning for Single Forward-pass Uncertainty. *ArXiv preprint*, abs/2102.11409. (Cited on pages 47, 113, 161, and 333)

Rob van der Goot. 2021. We Need to Talk about Train-dev-test Splits. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4485–4494. (Cited on page 71)

Liam van der Poel, Ryan Cotterell, and Clara Meister. 2022. Mutual Information Alleviates Hallucinations in Abstractive Summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5956–5965. (Cited on pages 4 and 68)

Daan van Esch, Tamar Lucassen, Sebastian Ruder, Isaac Caswell, and Clara Rivera. 2022. Writing System and Speaker Metadata for 2,800+ Language Varieties. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5035–5046. (Cited on page 6)

Marten Van Schijndel and Tal Linzen. 2018. Modeling Garden Path Effects without Explicit Hierarchical Syntax. In *CogSci*. (Cited on page 29)

Eva Vanmassenhove, Christian Hardmeier, and Andy Way. 2018. Getting Gender Right in Neural Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3003–3008. (Cited on page 4)

Tamás Váradi, Steven Krauwer, Peter Wittenburg, Martin Wynne, and Kimmo Koskenniemi. 2008. CLARIN: Common Language Resources and Technology Infrastructure. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. (Cited on page 77)

Giovanni Battista Varile, Ronald Cole, Ronald Allan Cole, Antonio Zampolli, Joseph Mariani, Hans Uszkoreit, and Annie Zaenen. 1997. Survey of the State of the Art in Human Language Technology. (Cited on page 27)

Neeraj Varshney and Chitta Baral. 2022. Model Cascading: Towards Jointly Improving Efficiency and Accuracy of NLP Systems. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11007–11021. (Cited on page 68)

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008. (Cited on pages 78, 110, 111, 173, and 338)

Artem Vazhentsev, Gleb Kuzmin, Artem Shelmanov, Akim Tsvigun, Evgenii Tsymbalov, Kirill Fedyanin, Maxim Panov, Alexander Panchenko, Gleb Gusev, Mikhail Burtsev, Manvel Avetisian, and Leonid Zhukov. 2022. Uncertainty Estimation of Transformer Predictions for Misclassification Detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8237–8252. (Cited on page 67)

Artem Vazhentsev, Gleb Kuzmin, Akim Tsvigun, Alexander Panchenko, Maxim Panov, Mikhail Burtsev, and Artem Shelmanov. 2023. Hybrid Uncertainty Quantification for Selective Text Classification in Ambiguous Tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11659–11681. (Cited on pages 67 and 164)

Raúl Vázquez, Hande Celikkanat, Dennis Ulmer, Jörg Tiedemann, Swabha Swayamdipta, Wilker Aziz, Barbara Plank, Joris Baan, and Marie-Catherine de Marneffe. 2024. Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024). In *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024)*. (Cited on page 165)

Pablo Villalobos, Jaime Sevilla, Lennart Heim, Tamay Besiroglu, Marius Hobbhahn, and Anson Ho. 2022. Will We Run out of Data? An Analysis of the Limits of Scaling Datasets in Machine Learning. *ArXiv preprint*, abs/2211.04325. (Cited on page 171)

Veronika Vincze. 2014. *Uncertainty Detection in Natural Language Texts*. Ph.D. thesis, Szegedi Tudomanyegyetem (Hungary). (Cited on pages 31 and 167)

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual Is not Enough: BERT for Finnish. *ArXiv preprint*, abs/1912.07076. (Cited on pages 114, 163, and 333)

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu

Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272. (Cited on page 325)

Kailas Vodrahalli, Tobias Gerstenberg, and James Y Zou. 2022. Uncalibrated Models Can Improve Human-AI Collaboration. *Advances in Neural Information Processing Systems*, 35:4004–4016. (Cited on page 66)

Leandro Von Werra, Lewis Tunstall, Abhishek Thakur, Sasha Luccioni, Tristan Thrush, Aleksandra Piktus, Felix Marty, Nazneen Rajani, Victor Mustar, and Helen Ngo. 2022. Evaluate & Evaluation on the Hub: Better Best Practices for Data and Model Measurements. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 128–136. (Cited on page 78)

Vladimir Vovk. 2012. Conditional Validity of Inductive Conformal Predictors. In *Asian conference on machine learning*, pages 475–490. PMLR. (Cited on page 166)

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. 2005. *Algorithmic Learning in a Random World*, volume 29. (Cited on pages 12, 39, 124, 166, and 338)

Yoav Wald, Amir Feder, Daniel Greenfeld, and Uri Shalit. 2021. On Calibration And Out-of-domain Generalization. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 2215–2227. (Cited on page 113)

Peter Walley. 1991. Statistical Reasoning with Imprecise Probabilities. (Cited on page 57)

Bin Wang, Tianrui Li, Zheng Yan, Guangquan Zhang, and Jie Lu. 2020a. DeepPipe: A Distribution-free Uncertainty Quantification Approach for Time Series Forecasting. *Neurocomputing*, 397:11–19. (Cited on page 6)

Cheng Wang, Carolin Lawrence, and Mathias Niepert. 2021a. Uncertainty Estimation and Calibration with Finite-state Probabilistic RNNs. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. (Cited on page 111)

Dan Wang and Yi Shang. 2014. A New Active Labeling Method for Deep Learning. In *2014 International joint conference on neural networks (IJCNN)*, pages 112–119. IEEE. (Cited on page 68)

Deng-Bao Wang, Lanqing Li, Peilin Zhao, Pheng-Ann Heng, and Min-Ling Zhang. 2023a. On the Pitfall of Mixup for Uncertainty Calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7609–7618. (Cited on page 38)

Hao Wang, Luxi He, Rui Gao, and Flavio Calmon. 2024a. Aleatoric and Epistemic Discrimination: Fundamental Limits of Fairness Interventions. *Advances in Neural Information Processing Systems*, 36. (Cited on page 67)

Kaizheng Wang, Keivan Shariatmadar, Shireen Kudukkil Manchingal, Fabio Cuzzolin, David Moens, and Hans Hallez. 2024b. CreINNs: Credal-set Interval Neural Networks for Uncertainty Estimation in Classification Tasks. *ArXiv preprint*, abs/2401.05043. (Cited on page 58)

Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun. 2019. Improving Back-translation with Uncertainty-based Confidence Estimation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 791–802. (Cited on page 143)

Shuo Wang, Zhaopeng Tu, Shuming Shi, and Yang Liu. 2020b. On the Inference Calibration of Neural Machine Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3070–3079. (Cited on page 59)

Xinpeng Wang and Barbara Plank. 2023. ACTOR: active learning with annotator-specific classification heads to embrace human label variation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2046–2052. Association for Computational Linguistics. (Cited on page 69)

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou.

2023b. Self-consistency Improves Chain-of-thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. (Cited on page 61)

Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. 2021b. Understanding how Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMAP, And PacMAP for Data Visualization. *Journal of Machine Learning Research*, 22:201:1–201:73. (Cited on page 320)

Yongguang Wang, Huobin Tan, and Shuzhen Yao. 2021c. Curved SDE-Net Leads to Better Generalization for Uncertainty Estimates of DNNs. In *International Conference on Artificial Neural Networks*, pages 248–259. Springer. (Cited on page 56)

Yongguang Wang and Shuzhen Yao. 2021. Neural Stochastic Differential Equations with Neural Processes Family Members for Uncertainty Estimation in Deep Learning. *Sensors*, 21(11):3708. (Cited on page 56)

Zeerak Waseem, Smarika Lulz, Joachim Bingel, and Isabelle Augenstein. 2021. Disembodied Machine Learning: On the Illusion of Objectivity in NLP. *ArXiv preprint*, abs/2101.11974. (Cited on pages 3, 74, and 82)

Ronald L. Wasserstein, Allen L. Schirm, and Nicole A. Lazar. 2019. Moving to a World Beyond “p <0.05”. (Cited on pages 81, 84, and 95)

David Watson, Joshua O’Hara, Niek Tax, Richard Mudd, and Ido Guy. 2024. Explaining Predictive Uncertainty with Information Theoretic Shapley Values. *Advances in Neural Information Processing Systems*, 36. (Cited on page 167)

Robert Watt. 2021. The Fantasy of Carbon Offsetting. *Environmental Politics*, 30(7):1069–1088. (Cited on page 327)

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, Denny Zhou, et al. 2022. Chain-of-thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, 35:24824–24837. (Cited on pages 143, 151, and 338)

Max Welling and Yee Whye Teh. 2011. Bayesian Learning via Stochastic Gradient Langevin Dynamics. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 681–688. (Cited on page 42)

Yeming Wen, Ghassen Jerfel, Rafael Muller, Michael W Dusenberry, Jasper Snoek, Balaji Lakshminarayanan, and Dustin Tran. 2020a. Improving Calibration of Batchensemble with Data Augmentation. *Workshop on Uncertainty and Ro-Bustness in Deep Learning*. (Cited on page 38)

Yeming Wen, Ghassen Jerfel, Rafael Muller, Michael W. Dusenberry, Jasper Snoek, Balaji Lakshminarayanan, and Dustin Tran. 2021. Combining Ensembles and Data Augmentation Can Harm Your Calibration. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. (Cited on page 38)

Yeming Wen, Dustin Tran, and Jimmy Ba. 2020b. Batchensemble: An Alternative Approach to Efficient Ensemble and Lifelong Learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. (Cited on page 46)

Jonathan Wenger, Hedvig Kjellström, and Rudolph Triebel. 2020. Non-parametric Calibration for Classification. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 178–190. (Cited on page 37)

Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. 2020. Hyperparameter Ensembles for Robustness and Uncertainty Quantification. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. (Cited on page 46)

Jennifer C. White and Ryan Cotterell. 2021. Examining the Inductive Bias of Neural Language Models with Artificial Languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 454–463. (Cited on page 74)

Andreas Widoff. 2022. Equivalence and Polyvalence: A Case for the Stratification of Semantics. *Public Journal of Semiotics*, 10(1):1–25. (Cited on page 30)

Martijn Wieling, Josine Rawee, and Gertjan van Noord. 2018. Reproducibility in Computational Linguistics: Are We Willing to Share? *Computational Linguistics*, 44(4):641–649. (Cited on page 77)

Norbert Wiener. 1949. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications*. (Cited on page 47)

Wikimedia Commons. 2022a. Iris Setosa. (Cited on page 50)

Wikimedia Commons. 2022b. Iris Versicolor. (Cited on page 50)

Wikimedia Commons. 2022c. Iris Virginica. (Cited on page 50)

Wikimedia Foundation. 2022. Wikimedia Downloads. `https://dumps.wikimedia.org/`. (Cited on page 129)

Wikipedia contributors. 2024. Wikipedia Statistics. `https://stats.wikimedia.org/EN/BotActivityMatrixCreates.htm`. [Online; accessed 12.04.2024]. (Cited on page 6)

Frank Wilcoxon. 1992. Individual Comparisons by Ranking Methods. In *Breakthroughs in Statistics*, pages 196–202. (Cited on pages 80 and 87)

Christopher K. I. Williams. 1998. Computation with Infinite Neural Networks. *Neural Computation*, 10(5):1203–1216. (Cited on page 47)

Christopher K.I. Williams and Carl Edward Rasmussen. 2006. *Gaussian Processes for Machine Learning*, volume 2. (Cited on page 47)

Timothy Williamson. 2002. *Vagueness*. (Cited on page 27)

Robin Willink and Rod White. 2011. Disentangling Classical and Bayesian Approaches to Uncertainty Analysis. *Measurement Standards Laboratory, PO Box*, 31310. (Cited on page 14)

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. 2016. Deep Kernel Learning. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, volume 51 of *JMLR Workshop and Conference Proceedings*, pages 370–378. (Cited on page 47)

Andrew Gordon Wilson and Pavel Izmailov. 2020. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. (Cited on pages 45, 46, and 110)

Lisa Wimmer, Yusuf Sale, Paul Hofman, Bernd Bischl, and Eyke Hüllermeier. 2023. Quantifying Aleatoric and Epistemic Uncertainty in Machine Learning: Are Conditional Entropy and Mutual Information Appropriate Measures? In *Uncertainty in Artificial Intelligence*, pages 2282–2292. PMLR. (Cited on page 167)

Jim Winkens, Rudy Bunel, Abhijit Guha Roy, Robert Stanforth, Vivek Natarajan, Joseph R. Ledsam, Patricia MacWilliams, Pushmeet Kohli, Alan Karthikesalingam, Simon Kohl, et al. 2020. Contrastive Training for Improved Out-of-distribution Detection. *ArXiv preprint*, abs/2007.05566. (Cited on page 113)

John Michael Winn. 2004. Variational Message Passing and Its Applications. (Cited on page 278)

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. (Cited on page 325)

Jae Oh Woo. 2022. Analytic Mutual Information in Bayesian Neural Networks. In *IEEE International Symposium on Information Theory, ISIT 2022, Espoo, Finland, June 26 - July 1, 2022*, pages 300–305. (Cited on page 54)

Ben Wu, Yue Li, Yida Mu, Carolina Scarton, Kalina Bontcheva, and Xingyi Song. 2023a. Don't waste a single annotation: Improving single-label classifiers through soft labels. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5347–5355. Association for Computational Linguistics. (Cited on page 63)

Xixin Wu and Mark Gales. 2021. Should Ensemble Members Be Calibrated? *ArXiv preprint*, abs/2101.05397. (Cited on page 38)

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023b. Fine-grained Human Feedback Gives Better Rewards For Language Model Training. In *Thirty-seventh Conference on Neural Information Processing Systems*. (Cited on page 143)

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex Machina: Personal Attacks Seen at Scale. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1391–1399. (Cited on page 96)

Alexandros Xenos, Themos Stafylakis, Ioannis Patras, and Georgios Tzimiropoulos. 2023. A Simple Baseline for Knowledge-based Visual Question Answering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 14871–14877. Association for Computational Linguistics. (Cited on page 77)

Tim Z. Xiao, Aidan N. Gomez, and Yarin Gal. 2020. Wat Zei Je? Detecting Out-of-distribution Translations with Variational Transformers. *ArXiv preprint*, abs/2006.08344. (Cited on pages 60, 111, and 334)

Yijun Xiao and William Yang Wang. 2021. On Hallucination and Predictive Uncertainty in Conditional Language Generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2734–2744. (Cited on pages 4, 67, and 68)

Yuxin Xiao, Paul Pu Liang, Umang Bhatt, Willie Neiswanger, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2022. Uncertainty Quantification with Pre-trained Language Models: A Large-scale Empirical Analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7273–7284. (Cited on page 59)

Chengxing Xie, Canyu Chen, Feiran Jia, Ziyu Ye, Kai Shu, Adel Bibi, Ziniu Hu, Philip Torr, Bernard Ghanem, and Guohao Li. 2024. Can Large Language Model Agents Simulate Human Trust Behaviors? *ArXiv preprint*, abs/2402.04559. (Cited on page 65)

Johnathan Xie, Annie S. Chen, Yoonho Lee, Eric Mitchell, and Chelsea Finn. Calibrating Language Models with Adaptive Temperature Scaling. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*. (Cited on page 60)

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An Explanation of In-context Learning as Implicit Bayesian Inference. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. (Cited on page 91)

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*. (Cited on pages 60 and 62)

Chen Xu and Yao Xie. 2021. Conformal Prediction Interval for Dynamic Time-series. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 11559–11569. (Cited on page 40)

Frank F. Xu, Uri Alon, and Graham Neubig. 2023a. Why do nearest neighbor language models work? In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 38325–38341. PMLR. (Cited on page 124)

Jiacheng Xu, Shrey Desai, and Greg Durrett. 2020. Understanding Neural Abstractive Summarization Models via Uncertainty. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6275–6281. (Cited on pages 4 and 167)

Winnie Xu, Ricky T. Q. Chen, Xuechen Li, and David Duvenaud. 2022. Infinitely Deep Bayesian Neural Networks with Stochastic Differential Equations. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, volume 151 of *Proceedings of Machine Learning Research*, pages 721–738. (Cited on page 56)

Yunpeng Xu, Wenge Guo, and Zhi Wei. 2023b. Conformal Risk Control for Ordinal Classification. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pages 2346–2355. (Cited on page 40)

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination Is Inevitable: An Innate Limitation of Large Language Models. *ArXiv preprint*, abs/2401.11817. (Cited on page 171)

Boyang Xue, Hongru Wang, Weichao Wang, Rui Wang, Sheng Wang, Zeming Liu, and Kam-Fai Wong. 2024a. A Comprehensive Study of Multilingual Confidence Estimation on Large Language Models. *ArXiv preprint*, abs/2402.13606. (Cited on pages 60 and 62)

Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. 2024b. To Repeat or not to Repeat: Insights from Scaling LLM under Token-crisis. *Advances in Neural Information Processing Systems*, 36. (Cited on page 172)

Adam Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. 2023. Bayesian Low-rank adaptation for Large Language Models. In *Socially Responsible Language Modelling Research*. (Cited on page 61)

Adam X. Yang, Maxime Robeyns, Thomas Coste, Jun Wang, Haitham Bou-Ammar, and Laurence Aitchison. 2024. Bayesian Reward Models for LLM Alignment. *ArXiv preprint*, abs/2402.13210. (Cited on page 63)

Jie Yang, Shuailong Liang, and Yue Zhang. 2018. Design Challenges and Misconceptions in Neural Sequence Labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3879–3889. (Cited on page 80)

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled Text Generation with Future Discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535. (Cited on pages 138 and 164)

Shingo Yashima, Teppei Suzuki, Kohta Ishikawa, Ikuro Sato, and Rei Kawakami. 2022. Feature Space Particle Inference for Neural Network Ensembles. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 25452–25468. (Cited on page 46)

Fanghua Ye, Mingming Yang, Jianhui Pang, Longyue Wang, Derek F Wong, Emine Yilmaz, Shuming Shi, and Zhaopeng Tu. 2024. Benchmarking LLMs via Uncertainty Quantification. *ArXiv preprint*, abs/2401.12794. (Cited on page 59)

Nanyang Ye, Kaican Li, Lanqing Hong, Haoyue Bai, Yiting Chen, Fengwei Zhou, and Zhenguo Li. 2021. OOD-Bench: Benchmarking and Understanding Out-of-distribution Generalization Datasets and Algorithms. *ArXiv preprint*, abs/2106.03721. (Cited on page 76)

Ming Yin, Jennifer Wortman Vaughan, and Hanna M. Wallach. 2019. Understanding the Effect of Accuracy on Trust in Machine Learning Models. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019*, page 279. (Cited on page 65)

Gal Yona, Roee Aharoni, and Mor Geva. 2024. Can Large Language Models Faithfully Express their Intrinsic Uncertainty in Words? *arXiv preprint arXiv: 2405.16908*. (Cited on page 62)

Kiyon Yoo, Jangho Kim, Jiho Jang, and Nojun Kwak. 2022. Detection of Adversarial Examples in Text Classification: Benchmark and Baseline via Robust Density Estimation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3656–3672. (Cited on page 164)

Hanlin Yu, Marcelo Hartmann, Bernardo Williams Moreno Sanchez, Mark Girolami, and Arto Klami. 2024. Riemannian laplace approximation with the fisher metric. In *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, volume 238 of *Proceedings of Machine Learning Research*, pages 820–828. PMLR. (Cited on page 45)

Ke-Hai Yuan and Kentaro Hayashi. 2003. Bootstrap Approach to Inference and Power Analysis Based on three Test Statistics for Covariance Structure Models. *British Journal of Mathematical and Statistical Psychology*, 56(1):93–110. (Cited on page 79)

Margaux Zaffran, Olivier Féron, Yannig Goude, Julie Josse, and Aymeric Dieuleveut. 2022. Adaptive Conformal Predictions for Time Series. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 25834–25866. (Cited on page 40)

Sheheryar Zaidi, Arber Zela, Thomas Elsken, Chris C. Holmes, Frank Hutter, and Yee Whye Teh. 2021. Neural Ensemble Search for Uncertainty Estimation and Dataset Shift. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 7898–7911. (Cited on page 46)

Marcos Zampieri, Preslav Nakov, and Yves Scherrer. 2020. Natural Language Processing for Similar Languages, Varieties, and Dialects: A Survey. *Natural Language Engineering*, 26(6):595–612. (Cited on page 6)

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent Neural Network Regularization. *arXiv preprint arXiv:1409.2329*. (Cited on page 332)

Günter Zech. 2002. Frequentist and Bayesian Confidence Intervals. *EPJ direct*, 4:1–81. (Cited on page 16)

Chrysoula Zerva, Taisiya Glushkova, Ricardo Rei, and André F.T. Martins. 2022. Better Uncertainty Quantification for Machine Translation Evaluation. *arXiv e-prints*, pages arXiv–2204. (Cited on page 143)

Yuanzhao Zhai, Han Zhang, Yu Lei, Yue Yu, Kele Xu, Dawei Feng, Bo Ding, and Huaimin Wang. 2024. Uncertainty-penalized Reinforcement Learning from Human Feedback with Diverse Reward LoRA Ensembles. *ArXiv preprint*, abs/2401.00243. (Cited on page 63)

Caiqi Zhang, Fangyu Liu, Marco Basaldella, and Nigel Collier. 2024a. LUQ: Long-text Uncertainty Quantification for LLMs. *ArXiv preprint*, abs/2403.20279. (Cited on page 60)

Chang-Bin Zhang, Peng-Tao Jiang, Qibin Hou, Yunchao Wei, Qi Han, Zhen Li, and Ming-Ming Cheng. 2021a. Delving Deep into Label Smoothing. *IEEE Transactions on Image Processing*, 30:5984–5996. (Cited on page 37)

Guodong Zhang, Shengyang Sun, David Duvenaud, and Roger B. Grosse. 2018a. Noisy Natural Gradient as Variational Inference. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5847–5856. (Cited on page 44)

Hanlin Zhang, Yifan Zhang, Yaodong Yu, Dhruv Madeka, Dean Foster, Eric P. Xing, Himabindu Lakkaraju, and Sham M. Kakade. A Study on the Calibration of In-context Learning. In *I Can't Believe It's Not Better Workshop: Failure Modes in the Age of Foundation Models*. (Cited on page 59)

Hanning Zhang, Shizhe Diao, Yong Lin, Yi R. Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2023a. R-Tuning: Teaching Large Language Models to Refuse Unknown Questions. *ArXiv preprint*, abs/2311.09677. (Cited on page 62)

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2018b. Mixup: Beyond Empirical Risk Minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. (Cited on page 38)

Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021b. Trading off Diversity and Quality in Natural Language Generation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 25–33. (Cited on page 123)

Jize Zhang, Bhavya Kailkhura, and Thomas Yong-Jin Han. 2020a. Mix-N-Match: Ensemble and Compositional Methods for Uncertainty Calibration in Deep Learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11117–11128. (Cited on pages 37 and 38)

Linjun Zhang, Zhun Deng, Kenji Kawaguchi, and James Zou. 2022a. When and how Mixup Improves Calibration. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 26135–26160. (Cited on page 38)

Mike Zhang and Barbara Plank. 2021. Cartography Active Learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 395–406. (Cited on page 69)

Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. 2020b. Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.* (Cited on page 43)

Shujian Zhang, Chengyue Gong, and Eunsol Choi. 2021c. Knowing more about Questions Can Help: Improving Calibration in Question Answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1958–1970. (Cited on page 59)

Shun Zhang, Zhenfang Chen, Sunli Chen, Yikang Shen, Zhiqing Sun, and Chuang Gan. 2024b. Improving Reinforcement Learning from Human Feedback with Efficient Reward Model Ensemble. *ArXiv preprint*, abs/2401.16635. (Cited on page 63)

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022b. OPT: Open Pre-trained Transformer Language Models. *ArXiv preprint*, abs/2205.01068. (Cited on page 129)

Tianhang Zhang, Lin Qiu, Qipeng Guo, Cheng Deng, Yue Zhang, Zheng Zhang, Chenghu Zhou, Xinbing Wang, and Luoyi Fu. 2023b. Enhancing Uncertainty-based Hallucination Detection with Stronger Focus. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP*

2023, Singapore, December 6-10, 2023, pages 915–932. Association for Computational Linguistics. (Cited on page 67)

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020c. BERTscore: Evaluating Text Generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.* (Cited on page 135)

Xiaoying Zhang, Jean-Francois Ton, Wei Shen, Hongning Wang, and Yang Liu. 2024c. Overcoming Reward Overoptimization via Adversarial Policy Optimization with Lightweight Uncertainty Estimation. *ArXiv preprint*, abs/2403.05171. (Cited on page 63)

Xinyu Zhang, Hanbin Hong, Yuan Hong, Peng Huang, Binghui Wang, Zhongjie Ba, and Kui Ren. 2023c. Text-CRS: A generalized certified robustness framework against textual adversarial attacks. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 53–53. IEEE Computer Society. (Cited on page 133)

Yunfeng Zhang, Q. Vera Liao, and Rachel K. E. Bellamy. 2020d. Effect of Confidence and Explanation on Accuracy and Trust Calibration in AI-Assisted Decision Making. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 295–305. (Cited on page 66)

Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2022c. A Survey of Active Learning for Natural Language Processing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6166–6190. (Cited on page 68)

Xujiang Zhao, Feng Chen, Shu Hu, and Jin-Hee Cho. 2020. Uncertainty Aware Semi-Supervised Learning on Graph Data. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.* (Cited on page 49)

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023.* (Cited on page 149)

Xin Zheng, Zhirui Zhang, Junliang Guo, Shujian Huang, Boxing Chen, Weihua Luo, and Jiajun Chen. 2021. Adaptive Nearest Neighbor Machine Translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 368–374. (Cited on page 124)

Zhuobin Zheng, Chun Yuan, Xinrui Zhu, Zhihui Lin, Yangyang Cheng, Cheng Shi, and Jiahui Ye. 2019. Self-supervised Mixture-of-experts by Uncertainty Estimation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 5933–5940. (Cited on page 68)

Kaitlyn Zhou, Jena D. Hwang, Xiang Ren, and Maarten Sap. 2024. Relying on the Unreliable: The Impact of Language Models' Reluctance to Express Uncertainty. *ArXiv preprint*, abs/2401.06730. (Cited on page 144)

Kaitlyn Zhou, Dan Jurafsky, and Tatsunori Hashimoto. 2023. Navigating the Grey Area: Expressions of Overconfidence and Uncertainty in Language Models. *arXiv e-prints*, pages arXiv–2302. (Cited on pages 30, 62, 153, and 169)

Xiang Zhou, Yixin Nie, and Mohit Bansal. 2022. Distributed NLI: Learning to Predict Human Opinion Distributions for Language Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 972–987. (Cited on page 63)

Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. 2002. Ensembling Neural Networks: Many Could Be Better than All. *Artificial intelligence*, 137(1-2):239–263. (Cited on page 46)

Chiwei Zhu, Benfeng Xu, Quan Wang, Yongdong Zhang, and Zhendong Mao. 2023. On the calibration of large language models and alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 9778–9795. Association for Computational Linguistics. (Cited on pages 37, 59, 63, and 172)

Daniel Zhu, Arnaud Martin, Yolande Le Gall, Jean-Christophe Dubois, and Vincent Lemaire. 2021. Evidential Nearest Neighbours in Active Learning. In *Worksop on Interactive Adaptive Learning (IAL)-ECML-PKDD*. (Cited on page 69)

Lingxue Zhu and Nikolay Laptev. 2017. Deep and Confident Prediction for Time Series at Uber. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 103–110. IEEE. (Cited on page 6)

Brian J. Zikmund-Fisher, Dylan M. Smith, Peter A. Ubel, and Angela Fagerlin. 2007. Validation of the Subjective Numeracy Scale: Effects of Low Numeracy on Comprehension of Risk Communications and Utility Elicitations. *Medical Decision Making*, 27(5):663–671. (Cited on page 66)

Steve Ziliak and Deirdre Nansen McCloskey. 2008. *The Cult of Statistical Significance: How the Standard Error Costs Us Jobs, Justice, and Lives.* (Cited on page 81)

Thomas Zollo, Todd Morrill, Zhun Deng, Jake Snell, Toniann Pitassi, and Richard Zemel. 2023. Prompt risk control: A rigorous framework for responsible deployment of large language models. In *Socially Responsible Language Modelling Research.* (Cited on pages 59, 164, and 165)

Shi Zong, Ashutosh Baheti, Wei Xu, and Alan Ritter. 2020. Extracting Covid-19 Events from Twitter. *ArXiv preprint*, abs/2006.02567. (Cited on page 77)

Shoshana Zuboff. 2023. The Age of Surveillance Capitalism. In *Social Theory Re-wired*, pages 203–213. (Cited on page 171)

Harun Zulić. 2019. How AI Can Change/Improve/Influence Music Composition, Performance and Education: Three Case Studies. In *INSAM Journal of Contemporary Music, Art and Technology*, 2, pages 100–114. (Cited on page 170)

A | Theoretical Appendix

"*Physics is searching for a* ***theory of everything****. Deep learning is searching for a* ***theory of anything****.*"

—Zachary Lipton on Twitter.

Thesis	Appendix
Section 2.1.2	Appendix A.1
Section 2.2.3	Appendices A.2, A.3, A.5 and A.6
Section 4.1.1	Appendix A.7
Section 4.1.3	Appendices A.9 and A.10
Section 4.1.4	Appendices A.6 and A.11 to A.13

Table A.1: Correspondences between sections of the theoretical appendix and thesis chapters.

This appendix contains additional derivations and proofs for some of the main chapters in this thesis. Table A.1 gives an overview over the correspondences between thesis chapters and sections in this appendix.

A.1 Relationship between Beta and Gamma function

Here, we further elaborate on the connection between the Beta and the Gamma function, used to derive the predictive prior and posterior distribution of a Beta distribution with Bernoulli likelihood in Equations (2.28) and (2.34). The Beta function is commonly defined in terms of Gamma functions, namely

$$B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}, \tag{A.1}$$

and recall the definition of the Gamma function as

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1}\exp(-x)\mathrm{d}x. \tag{A.2}$$

Alternatively, the Beta function can be stated as

$$\mathrm{B}(\alpha, \beta) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1}\mathrm{d}x. \tag{A.3}$$

This connection arises by evaluating the following product:

$$\Gamma(\alpha)\Gamma(\beta) = \Big(\int_0^\infty x^{\alpha-1}\exp(-x)\mathrm{d}x\Big)\Big(\int_0^\infty y^{\beta-1}\exp(-y)\mathrm{d}y\Big) \tag{A.4}$$

$$= \int_0^\infty \int_0^\infty x^{\alpha-1}y^{\beta-1}\exp\big(-(x+y)\big)\mathrm{d}x\mathrm{d}y. \tag{A.5}$$

In order to simplify the integration, we apply a change of variables by substituting $x = uv$ and $y = u(1-v)$. To account for the change of variables during the integration, we also need to evaluate the determinant of the Jacobian as

$$|\mathbf{J}| = \begin{vmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{vmatrix} = \begin{vmatrix} v & u \\ 1-v & -u \end{vmatrix} = -uv - u(1-v) = -u. \tag{A.6}$$

By writing u and v in terms of x and y, we obtain that $u = x+y$ and $v = x/(x+y)$, which implies that the limits for the integration remain 0 to ∞ for u and become 0 to 1 for v. Using all of these insights, we can now show that

$$\Gamma(\alpha)\Gamma(\beta) = \int_0^\infty \int_0^\infty x^{\alpha-1}y^{\beta-1}\exp\big(-(x+y)\big)\mathrm{d}x\mathrm{d}v \tag{A.7}$$

$$= \int_0^1 \int_0^\infty (uv)^{\alpha-1}\big(u(1-v)\big)^{\beta-1} \exp\big(-(uv+u(1-v))\big)|-u|\mathrm{d}u\mathrm{d}v \tag{A.8}$$

$$= \int_0^1 \int_0^\infty u^{\alpha-1}v^{\alpha-1}u^{\beta-1}(1-v)^{\beta-1}\exp(-u)u\mathrm{d}u\mathrm{d}v \tag{A.9}$$

$$= \int_0^1 \int_0^\infty u^{\alpha+\beta-1}v^{\alpha-1}u^{\beta-1}(1-v)^{\beta-1}\exp(-u)\mathrm{d}u\mathrm{d}v \tag{A.10}$$

$$= \Big(\int_0^1 v^{\alpha-1}(1-v)^{\beta-1}\mathrm{d}v\Big)\Big(\int_0^\infty u^{\alpha+\beta-1}\exp(-u)\mathrm{d}u\Big) \tag{A.11}$$

$$= B(\alpha, \beta)\Gamma(\alpha+\beta), \tag{A.12}$$

from which the connection between the two definition follows.

A.2 Expectation of the Dirichlet Distribution

Here, we show results for the quantities $\mathbb{E}[\pi_k]$ and $\mathbb{E}[\log \pi_k]$ that appear in Section 2.2.3. For the first, we follow the derivation by Miller (2011). Another proof is given by Lin (2016).

$$\mathbb{E}[\pi_k] = \int \cdots \int \pi_k \frac{\Gamma(\alpha_0)}{\prod_{k'=1}^{K} \Gamma(\alpha'_k)} \prod_{k'=1}^{K} \pi_{k'}^{\alpha_{k'}-1} \mathrm{d}\pi_1 \ldots \mathrm{d}\pi_K. \tag{A.13}$$

Moving $\pi_k^{\alpha_k - 1}$ out of the product:

$$= \int \cdots \int \frac{\Gamma(\alpha_0)}{\prod_{k'=1}^{K} \Gamma(\alpha_{k'})} \pi_k^{\alpha_k - 1 + 1} \prod_{k' \neq k} \pi_{k'}^{\alpha_{k'}-1} \mathrm{d}\pi_1 \ldots \mathrm{d}\pi_K. \tag{A.14}$$

For the next step, we define a new set of Dirichlet parameters with $\beta_k = \alpha_k + 1$ and $\forall k' \neq k : \beta_{k'} = \alpha_{k'}$. For those new parameters, $\beta_0 = \sum_k \beta_k = 1 + \alpha_0$. So by virtue of the Gamma function's property that $\Gamma(\beta_0) = \Gamma(\alpha_0 + 1) = \alpha_0 \Gamma(\alpha_0)$, replacing all terms in the normalization factor yields

$$= \int \cdots \int \frac{\alpha_k}{\alpha_0} \frac{\Gamma(\beta_0)}{\prod_{k'=1}^{K} \Gamma(\beta_{k'})} \prod_{k'=1}^{K} \pi_{k'}^{\beta_{k'}-1} \mathrm{d}\pi_1 \ldots \mathrm{d}\pi_K = \frac{\alpha_k}{\alpha_0}, \tag{A.15}$$

where in the last step we obtain the final result, since the Dirichlet with new parameters β_k must nevertheless integrate to 1, and the integrals do not regard α_k or α_0. For the expectation $\mathbb{E}[\log \pi_k]$, we first rephrase the Dirichlet distribution in terms of the exponential families (Kupperman, 1964). The exponential families encompass many commonly-used distributions, such as the normal, exponential, Beta or Poisson, which all follow the form

$$p(\mathbf{x}; \boldsymbol{\eta}) = h(\mathbf{x}) \exp\big(\boldsymbol{\eta}^{\mathrm{T}} u(\mathbf{x}) - A(\boldsymbol{\eta})\big), \tag{A.16}$$

with *natural parameters* $\boldsymbol{\eta}$, *sufficient statistic* $u(\mathbf{x})$, and *log-partition function* $A(\boldsymbol{\eta})$. For the Dirichlet distribution, Winn (2004) provides the sufficient statistic as $u(\boldsymbol{\pi}) = [\log \boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_K]^T$ and the log-partition function

$$A(\boldsymbol{\alpha}) = \sum_{k=1}^{K} \log \Gamma(\alpha_k) - \log \Gamma(\alpha_0). \tag{A.17}$$

By Mao (2019), we also find that by the moment-generating function that for the sufficient statistic, its expectation can be derived by

$$\mathbb{E}[u(\mathbf{x})_k] = \frac{\partial A(\boldsymbol{\eta})}{\partial \eta_k}. \tag{A.18}$$

Therefore, we can evaluate the expected value of $\log \pi_k$ (i.e. the sufficient statistic) by inserting the definition of the log-partition function in Equation (A.17) into Equation (A.18):

$$\mathbb{E}[\log \pi_k] = \frac{\partial}{\partial \alpha_k} \sum_{k=1}^{K} \log \Gamma(\alpha_k) - \log \Gamma(\alpha_0) = \psi(\alpha_k) - \psi(\alpha_0), \tag{A.19}$$

which corresponds precisely to the definition of the digamma function as $\psi(x) = \frac{d}{dx} \log \Gamma(x)$.

A.3 Entropy of the Dirichlet Distribution

The following derivation for the entropy of the Dirichlet which appears in Section 2.2.3 is adapted from Lin (2016), with the result stated in Charpentier et al. (2020) as well.

$$\begin{aligned}
\mathrm{H}[p(\boldsymbol{\pi} \mid \boldsymbol{\alpha})] &= -\mathbb{E}[\log p(\boldsymbol{\pi} \mid \boldsymbol{\alpha})] && \text{(A.20)}\\
&= -\mathbb{E}\Big[\log \Big(\frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}\Big)\Big] && \text{(A.21)}\\
&= -\mathbb{E}\Big[-\log \mathrm{B}(\boldsymbol{\alpha}) + \sum_{k=1}^{K} (\alpha_k - 1) \log \pi_k\Big] && \text{(A.22)}\\
&= \log \mathrm{B}(\boldsymbol{\alpha}) - \sum_{k=1}^{K} (\alpha_k - 1) \mathbb{E}[\log \pi_k]. && \text{(A.23)}
\end{aligned}$$

Using Equation (A.19):

$$\begin{aligned}
&= \log \mathrm{B}(\boldsymbol{\alpha}) - \sum_{k=1}^{K} (\alpha_k - 1)\big(\psi(\alpha_k) - \psi(\alpha_0)\big) && \text{(A.24)}\\
&= \log \mathrm{B}(\boldsymbol{\alpha}) + \sum_{k=1}^{K} (\alpha_k - 1)\psi(\alpha_0) - \sum_{k=1}^{K} (\alpha_k - 1)\psi(\alpha_k) && \text{(A.25)}\\
&= \log \mathrm{B}(\boldsymbol{\alpha}) + (\alpha_0 - K)\psi(\alpha_0) - \sum_{k=1}^{K} (\alpha_k - 1)\psi(\alpha_k). && \text{(A.26)}
\end{aligned}$$

A.4 Expected Entropy of the Dirichlet Distribution

The following derivation for the expected entropy of the Dirichlet which appears in Section 2.2.3 is adapted from Malinin and Gales (2018) appendix section C.4. In the following, we assume that $\forall k \in \mathbb{K} : \pi_k > 0$:

$$\mathbb{E}_{p(\boldsymbol{\pi}|\mathbf{x},\hat{\boldsymbol{\theta}})}\Big[\mathrm{H}\big[P(y \mid \boldsymbol{\pi})\big]\Big] = \int p(\boldsymbol{\pi} \mid \mathbf{x}, \hat{\boldsymbol{\theta}})\Big(-\sum_{k=1}^{K} \pi_k \log \pi_k\Big)\mathrm{d}\boldsymbol{\pi} \tag{A.27}$$

$$= -\sum_{k=1}^{K} \int p(\boldsymbol{\pi} \mid \mathbf{x}, \hat{\boldsymbol{\theta}})\big(\pi_k \log \pi_k\big)\mathrm{d}\boldsymbol{\pi}. \tag{A.28}$$

Inserting the definition of $p(\boldsymbol{\pi}| \mathbf{x}, \hat{\boldsymbol{\theta}}) \approx p(\boldsymbol{\pi} \mid \mathbf{x}, \mathbb{D})$:

$$= -\sum_{k=1}^{K}\left(\frac{\Gamma(\alpha_0)}{\prod_{k'=1}^{K}\Gamma(\alpha_{k'})}\int \pi_k \log \pi_k \prod_{k'=1}^{K} \pi_{k'}^{\alpha_{k'}-1}\mathrm{d}\boldsymbol{\pi}\right). \tag{A.29}$$

Singling out the factor π_k:

$$= -\sum_{k=1}^{K}\left(\frac{\Gamma(\alpha_0)}{\Gamma(\alpha_k)\prod_{k'\neq k}\Gamma(\alpha_{k'})}\pi_k^{\alpha_k-1}\int \pi_k \log \pi_k \prod_{k'\neq k} \pi_{k'}^{\alpha_{k'}-1}\mathrm{d}\boldsymbol{\pi}\right). \tag{A.30}$$

Adjusting the normalizing constant (this is the same trick used in Appendix A.2):

$$= -\sum_{k=1}^{K}\left(\frac{\alpha_k}{\alpha_0}\int \frac{\Gamma(\alpha_0+1)}{\Gamma(\alpha_k+1)\prod_{k'\neq k}\Gamma(\alpha_{k'})}\pi_k^{\alpha_k-1} \log \pi_k \prod_{k'\neq k} \pi_{k'}^{\alpha_{k'}-1}\mathrm{d}\boldsymbol{\pi}\right). \tag{A.31}$$

Using the identity $\mathbb{E}[\log \pi_k] = \psi(\alpha_k) - \psi(\alpha_0)$ (Equation (A.19)). Since the expectation here is w.r.t. to a Dirichlet with concentration parameters $\alpha_k + 1$, we obtain

$$= -\sum_{k=1}^{K}\frac{\alpha_k}{\alpha_0}\Big(\psi(\alpha_k+1) - \psi(\alpha_0+1)\Big). \tag{A.32}$$

A.5 Kullback-Leibler Divergence between two Dirichlets

The following result appearing in Section 2.2.3 is presented using an adapted derivation by Lin (2016) and appears in Chen et al. (2018) and Joo et al. (2020) as a starting point for their variational objective. In the following we use $\text{Dir}(\boldsymbol{\pi}; \boldsymbol{\alpha})$ to denote distribution to be optimized, and $\text{Dir}(\boldsymbol{\pi}; \boldsymbol{\gamma})$ for the reference or target distribution.

$$\begin{aligned}
&\text{KL}\big[p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}) \,\big|\big|\, p(\boldsymbol{\pi} \mid \boldsymbol{\gamma})\big] \\
&= \mathbb{E}\Big[\log \frac{p(\boldsymbol{\pi} \mid \boldsymbol{\alpha})}{p(\boldsymbol{\pi} \mid \boldsymbol{\gamma})}\Big] = \mathbb{E}\big[\log p(\boldsymbol{\pi} \mid \boldsymbol{\alpha})\big] - \mathbb{E}\big[\log p(\boldsymbol{\pi} \mid \boldsymbol{\gamma})\big] \qquad \text{(A.33)} \\
&= \mathbb{E}\Big[-\log \text{B}(\boldsymbol{\alpha}) + \sum_{k=1}^{K} (\alpha_k - 1) \log \pi_k\Big] \\
&- \mathbb{E}\Big[-\log \text{B}(\boldsymbol{\gamma}) + \sum_{k=1}^{K} (\gamma_k - 1) \log \pi_k\Big]. \qquad \text{(A.34)}
\end{aligned}$$

Distributing and pulling out $\text{B}(\boldsymbol{\alpha})$ and $\text{B}(\boldsymbol{\gamma})$ out of the expectation (they don't depend on $\boldsymbol{\pi}$):

$$= -\log \frac{\text{B}(\boldsymbol{\gamma})}{\text{B}(\boldsymbol{\alpha})} + \mathbb{E}\Big[\sum_{k=1}^{K} (\alpha_k - 1) \log \pi_k - (\gamma_k - 1) \log \pi_k\Big] \qquad \text{(A.35)}$$

$$= -\log \frac{\text{B}(\boldsymbol{\gamma})}{\text{B}(\boldsymbol{\alpha})} + \mathbb{E}\Big[\sum_{k=1}^{K} (\alpha_k - \gamma_k) \log \pi_k\Big]. \qquad \text{(A.36)}$$

Moving the expectation inward and using the identity $\mathbb{E}[\pi_k] = \psi(\alpha_k) - \psi(\alpha_0)$ from Appendix A.2:

$$= -\log \frac{\text{B}(\boldsymbol{\gamma})}{\text{B}(\boldsymbol{\alpha})} + \sum_{k=1}^{K} (\alpha_k - \gamma_k)\big(\psi(\alpha_k) - \psi(\alpha_0)\big). \qquad \text{(A.37)}$$

The KL divergence is also used by some works as regularizer by penalizing the distance to a uniform Dirichlet with $\boldsymbol{\gamma} = \mathbf{1}$ (Sensoy et al., 2018). In this case, the result above can be derived to be

$$\text{KL}\big[p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}) \,\big|\big|\, p(\boldsymbol{\pi} \mid \mathbf{1})\big] = \log \frac{\Gamma(K)}{\text{B}(\boldsymbol{\alpha})} + \sum_{k=1}^{K} (\alpha_k - 1)\big(\psi(\alpha_k) - \psi(\alpha_0)\big), \qquad \text{(A.38)}$$

where the $\log \Gamma(K)$ term can also be omitted for optimization purposes, since it does not depend on $\boldsymbol{\alpha}$.

A.6 Mutual Information for Dirichlet Networks

As stated in Section 2.2.3, mutual information is a measure of distributional uncertainty in Dirichlet networks. To derive its closed-form expression, we start from Equation (2.73):

$$\mathrm{I}\Big[y, \boldsymbol{\pi} \;\Big|\; \mathbf{x}, \mathbb{D}\Big] = \mathrm{H}\Big[\mathbb{E}_{p(\boldsymbol{\pi}|\mathbf{x},\mathbb{D})}\big[P(y \mid \boldsymbol{\pi})\big]\Big] - \mathbb{E}_{p(\boldsymbol{\pi}|\mathbf{x},\mathbb{D})}\Big[\mathrm{H}\big[P(y \mid \boldsymbol{\pi})\big]\Big]. \tag{A.39}$$

Given that $\mathbb{E}[\pi_k] = \frac{\alpha_k}{\alpha_0}$ (Appendix A.2) and assuming that point estimate $p(\boldsymbol{\pi} \mid \mathbf{x}, \mathbb{D}) \approx p(\boldsymbol{\pi} \mid \mathbf{x}, \hat{\boldsymbol{\theta}})$ is sufficient (Malinin and Gales, 2018), we can identify the first term as the Shannon entropy $-\sum_{k=1}^{K} \pi_k \log \pi_k = -\sum_{k=1}^{K} \frac{\alpha_k}{\alpha_0} \log \frac{\alpha_k}{\alpha_0}$. Furthermore, the second part we already derived in Appendix A.4, and thus we obtain:

$$= -\sum_{k=1}^{K} \frac{\alpha_k}{\alpha_0} \log \frac{\alpha_k}{\alpha_0} + \sum_{k=1}^{K} \frac{\alpha_k}{\alpha_0}\Big(\psi(\alpha_k + 1) - \psi(\alpha_0 + 1)\Big) \tag{A.40}$$

$$= -\sum_{k=1}^{K} \frac{\alpha_k}{\alpha_0}\Big(\log \frac{\alpha_k}{\alpha_0} - \psi(\alpha_k + 1) + \psi(\alpha_0 + 1)\Big). \tag{A.41}$$

A.7 Connection between Softmax and Sigmoid

In this section we briefly outline the connection between the softmax and the sigmoid function, in order to show the applicability of results in Section 4.1 to both binary and multi-class classification problems. This connection was originally shown in Bridle (1990). Let the sigmoid function be defined as

$$\sigma(x) = \frac{\exp(x)}{1 + \exp(x)}, \tag{A.42}$$

and softmax according to the definition in Equation (0.2). The output of $f_{\boldsymbol{\theta}}$ in a multi-class classification problem with K classes corresponds to a K-dimensional column vector that is based on an affine transformation of the network's last intermediate hidden representation $\mathbf{x}_L$, such that $f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{W}_L \mathbf{x}_L$.[85] Correspondingly, the output of $f_{\boldsymbol{\theta}}$ for a single class c can be written as the dot product between $\mathbf{x}_L$ and the corresponding row vector of $\mathbf{W}_L$ denoted as $\mathbf{w}_L^{(c)}$, such that $f_{\boldsymbol{\theta}}(\mathbf{x})_k \equiv \mathbf{w}_L^{(k)T} \mathbf{x}_L$. For a classification problem

[85] The bias term $\mathbf{b}_L$ was omitted here for clarity.

with $K = 2$ classes, we can now rewrite the softmax probabilities in the following way:[86]

$$P_{\boldsymbol{\theta}}(y = 1 \mid \mathbf{x}) = \frac{\exp(\mathbf{w}_L^{(1)\mathrm{T}}\,\mathbf{x}_L)}{\exp(\mathbf{w}_L^{(0)\mathrm{T}}\,\mathbf{x}_L) + \exp(\mathbf{w}_L^{(1)\mathrm{T}}\,\mathbf{x}_L)}. \tag{A.43}$$

Subtracting a constant from the weight term inside the exponential function does not change the output of the softmax function. Using this property, we can show the sigmoid function to be a special case of the softmax for binary classification:

$$P_{\boldsymbol{\theta}}(y = 1 \mid \mathbf{x}) = \frac{\exp((\mathbf{w}_L^{(1)} - \mathbf{w}_L^{(0)})^{\mathrm{T}}\,\mathbf{x}_L)}{\exp((\mathbf{w}_L^{(0)} - \mathbf{w}_L^{(0)})^{\mathrm{T}}\,\mathbf{x}_L) + \exp((\mathbf{w}_L^{(1)} - \mathbf{w}_L^{(0)})^{\mathrm{T}}\,\mathbf{x}_L)} \tag{A.44}$$

$$= \frac{\exp((\mathbf{w}_L^{(1)} - \mathbf{w}_L^{(0)})^{\mathrm{T}}\,\mathbf{x}_L)}{1 + \exp((\mathbf{w}_L^{(1)} - \mathbf{w}_L^{(0)})^{\mathrm{T}}\,\mathbf{x}_L} = \frac{\exp(\mathbf{w}_L^{*\,\mathrm{T}}\,\mathbf{x}_L)}{1 + \exp(\mathbf{w}_L^{*\,\mathrm{T}}\,\mathbf{x}_L)}, \tag{A.45}$$

where $\mathbf{w}_L^* = \mathbf{w}_L^{(1)} - \mathbf{w}_L^{(0)}$ corresponds to the new parameter vector which is used to parametrize a single output unit for a network in the binary classification setting.

A.8 Construction of Polytopal Regions

In this section, we reiterate the reasoning by Hein et al. (2019) behind the construction the polytopal regions mentioned in Section 4.1.3. For this purpose, the authors define an additional diagonal matrix $\Delta_l(\mathbf{x})$ per layer l:

$$\Delta_l(\mathbf{x}) = \begin{bmatrix} \operatorname{sign}(f_{\boldsymbol{\theta}}^l(\mathbf{x})_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \operatorname{sign}(f_{\boldsymbol{\theta}}^l(\mathbf{x})_{n_l}) \end{bmatrix}. \tag{A.46}$$

Together with the linearization of the network at $\mathbf{x}$ explained in Equation (4.13), this is used to define a set of half-spaces for every neuron in the network:

$$\mathbb{H}_{l,i}(\mathbf{x}) = \big\{\, \mathbf{z} \in \mathbb{R}^d \;\big|\; \Delta_l(\mathbf{x})\big(\mathbf{V}_l(\mathbf{x})_i\,\mathbf{z} + \mathbf{a}_l(\mathbf{x})_i\big) \geq 0 \big\}. \tag{A.47}$$

Here, $\mathbf{V}_l(\mathbf{x})_i$ and $\mathbf{b}_l(\mathbf{x})_i$ denote the parts of the affine transformation obtained for the i-th neuron of the l-th layer, so the i-th row

[86] The following argument holds without loss of generality for $P_{\boldsymbol{\theta}}(y = 0 \mid \mathbf{x})$.

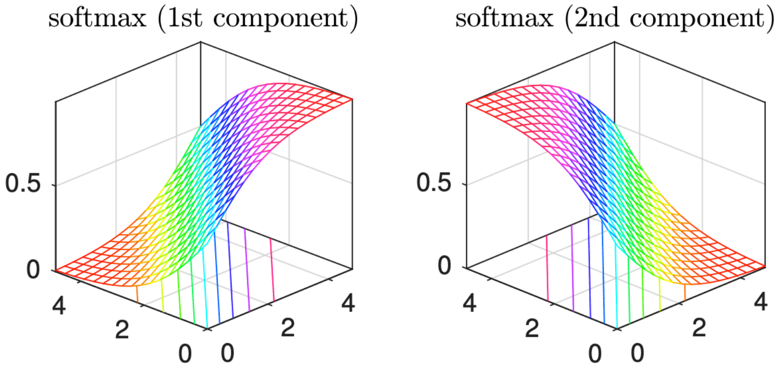


Figure A.1: Illustration taken from the work of Gao and Pavel (2017), illustrating the interplay of softmax probabilities between components for $K = 2$ in $\mathbb{R}^2$.

vector in $\mathbf{V}_l(\mathbf{x})$ and the i-th scalar in $\mathbf{b}_l(\mathbf{x})$, respectively. Finally, the polytope Q containing $\mathbf{x}$ is obtained by taking the intersection of all half-spaces induced by every neuron in the network:

$$Q(\mathbf{x}) = \bigcap_{l \in 1,\dots,L} \bigcap_{i \in 1,\dots,n_l} \mathbb{H}_{l,i}(\mathbf{x}). \tag{A.48}$$

A.9 Proof of Proposition 1

This section provides the proof of Proposition 1 in Section 4.1.3. We proceed to analyze the behavior of gradients in the limit via two more lemmas; First, we establish the saturating property of the softmax in Lemma 5, i.e. the model doesn't change its decision anymore in the limit.

Lemma 5. *Let $k, k' \in [K]$ be two arbitrary classes. It then holds for their corresponding output components (logits) that*

$$\lim_{f_{\boldsymbol{\theta}}(\mathbf{x})_k \to \pm\infty} \frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{k'}} \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_k = 0. \tag{A.49}$$

Proof. Here, we first begin by evaluating the derivative of one component of the function w.r.t. to an arbitrary component:

$$\frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{k'}} \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_k = \frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{k'}} \frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_k)}{\sum_{k'' \in [K]} \exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{k''})} \tag{A.50}$$

$$= \frac{\mathbb{1}\left(k = k'\right) \exp(f_{\boldsymbol{\theta}}(\mathbf{x})_k)}{\sum_{k'' \in [K]} \exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{k''})} - \frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_k) \exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{k'})}{\left(\sum_{k'' \in [K]} \exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{k''})\right)^2}. \tag{A.51}$$

This implies that

$$\frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{k'}}\bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_k = \begin{cases} -\frac{\exp(2f_{\boldsymbol{\theta}}(\mathbf{x})_k)}{\left(\sum_{k''\in[K]}\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{k''})\right)^2} + \frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_k)}{\sum_{k''\in[K]}\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{k''})} & \text{if } k = k' \\ -\frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_k + f_{\boldsymbol{\theta}}(\mathbf{x})_{k'})}{\left(\sum_{k''\in[K]}\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{k''})\right)^2} & \text{if } k \neq k' \end{cases} \tag{A.52}$$

or more compactly:

$$\frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{k'}}\bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_k = \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_k\big(\mathbb{1}\big(k = k'\big) - \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_{k'}\big).$$

Based on Equation (A.52), we can now investigate the asymptotic behavior for $f_{\boldsymbol{\theta}}(\mathbf{x})_k \to \infty$ more easily, starting with the $k = k'$ case:

$$\begin{aligned} &\lim_{f_{\boldsymbol{\theta}}(\mathbf{x})_k\to\infty} \frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{k'}}\bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_k \\ &= \underbrace{\lim_{f_{\boldsymbol{\theta}}(\mathbf{x})_k\to\infty} -\frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_k)}{\sum_{k''\in[K]}\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{k''})}\frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_k)}{\sum_{k''\in[K]}\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{k''})}}_{\text{-1}} \\ &+ \underbrace{\lim_{f_{\boldsymbol{\theta}}(\mathbf{x})_k\to\infty} \frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_k)}{\sum_{k''\in[K]}\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{k''})}}_{1} = 0. \end{aligned} \tag{A.53}$$

With the numerator and denominator being dominated by the exponentiated $f_{\boldsymbol{\theta}}(\mathbf{x})_k$ in Equation (A.53), the first term will tend to -1, while the second term will tend to 1, resulting in a derivative of 0. The case $k \neq k'$ can be analyzed the following way:

$$\begin{aligned} &\lim_{f_{\boldsymbol{\theta}}(\mathbf{x})_k\to\infty} \frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{k'}}\bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_k = \\ &\lim_{f_{\boldsymbol{\theta}}(\mathbf{x})_k\to\infty} \underbrace{\left(-\frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_k)}{\sum_{k''\in[K]}\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{k''})}\right)}_{-1}\underbrace{\left(\frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{k'})}{\sum_{k''\in[K]}\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{k''})}\right)}_{0} = 0. \end{aligned} \tag{A.54}$$

Again, we factorize the fraction in Equation (A.54) into the product of two softmax functions, one for component k, one for k'. The first factor will again tend to -1 as in the other case, however the second will approach 0, as only the sum in the denominator will approach infinity. As the limit of a product is the products of its limits, this lets the whole expression approach 0 in the limit. When $f_{\boldsymbol{\theta}}(\mathbf{x})_k \to -\infty$, both cases approach 0 due to the exponential function, which proves the lemma. □

How the interplay between different softmax components produces zero gradients in the limit is illustrated in Figure A.1. In Lemma 6, we compare the rate of growth of different components of $P_{\boldsymbol{\theta}}$. We show that for the decomposed function $P_{\boldsymbol{\theta}}$, the rate at which the softmax function converges to its output distribution in the limit outpaces the change in the underlying logits w.r.t. the network input.

Lemma 6. *Suppose that $f_{\boldsymbol{\theta}}$ is a ReLU-network. Let $\mathbf{x}' \in \mathbb{R}^D$, suppose $\boldsymbol{\alpha}$ is a scaling vector and that the associated PUP $\mathbb{P}(\mathbf{x}', d)$ has a corresponding matrix $\mathbf{V}$ with no zero entries. Then it holds for all $k' \in [K]$ that*

$$\lim_{\alpha_d \to \infty} \left(\frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{k'}} \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_k \right)^{-1} \Bigg|_{\mathbf{x}=\boldsymbol{\alpha}\circ\mathbf{x}'} - \left(\frac{\partial}{\partial x_d} f_{\boldsymbol{\theta}}(\mathbf{x})_{k'} \right) \Bigg|_{\mathbf{x}=\boldsymbol{\alpha}\circ\mathbf{x}'} = \infty. \tag{A.55}$$

Proof. We evaluate the first term of Equation (A.55) to show that it grows exponentially in the limit. By Lemma 2, we know that in the limit $\alpha_d \to \infty$ the vector $\boldsymbol{\alpha} \circ \mathbf{x}'$ will remain within $\mathbb{P}(\mathbf{x}', d)$. Since the matrix associated with this PUP has no zero entries, we know by Lemma 1 that the gradient of $f_{\boldsymbol{\theta}}(\mathbf{x})_k$ on dimension d is either always positive or negative, hence $f_{\boldsymbol{\theta}}(\mathbf{x})_k \to \pm\infty$. Given Lemma 5 describing the asymptotic behavior in the limit, it follows that

$$\lim_{f_{\boldsymbol{\theta}}(\mathbf{x})_k \to \pm\infty} \left(\frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{k'}} \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_k \right)^{-1} = \infty, \tag{A.56}$$

where we can see that the result is a symmetrical function displaying exponential growth in the limit of $f_{\boldsymbol{\theta}}(\mathbf{x})_k \to \pm\infty$. We now show that because we assumed $f_{\boldsymbol{\theta}}$ to be a neural network consisting of L affine transformations with ReLU activation functions, the output of the final layer is only going to be a linear combination of its inputs.[87] This can be proven by induction. Let us first look at the base case $L = 1$. In the rest of this proof, we denote $\mathbf{x}_l$ as the input to layer l, with $\mathbf{x}_1 \equiv \mathbf{x}$, and $\mathbf{W}_l, \mathbf{b}_l$ the corresponding layer parameters. $\mathbf{a}_l$ signifies the result of the affine transformation that is then fed into the activation function.

$$\begin{aligned}
f_{\boldsymbol{\theta}}(\mathbf{x}) &= \phi(\mathbf{a}_1) = \phi(\mathbf{W}_1 \mathbf{x}_1 + \mathbf{b}_1) \\
\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{x})}{\partial \mathbf{x}_1} &= \frac{\partial \phi(\mathbf{a}_1)}{\partial \mathbf{a}_1} \frac{\partial \mathbf{a}_1}{\partial \mathbf{x}_1} = \mathbf{1}(\mathbf{x}_1 > \mathbf{0})^{\mathrm{T}} \mathbf{W}_1 \\
\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_{1d}} &= \mathbb{1}\big(x_d > 0\big) w_{1d},
\end{aligned} \tag{A.57}$$

where $\mathbf{1}(\mathbf{x}_1 > \mathbf{0}) = [\mathbb{1}\big(x_{11} > 0\big), \ldots, \mathbb{1}\big(x_{1d} > 0\big)]^{\mathrm{T}}$, w_{1d} denoting the d-th column of $\mathbf{W}_1$. This is a linear function, which proves the

[87] Here we make the argument for the whole function $f_{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}^K$, but the conclusions also applies to every output component of the function $f_{\boldsymbol{\theta}}(\mathbf{x})_k$.

base case. Let now $\frac{\partial \mathbf{x}_l}{\partial \mathbf{x}_1}$ denote the partial derivative of the input to the l-th layer w.r.t. to the input and suppose that it is linear by the inductive hypothesis. Augmenting the corresponding network by another linear adds another term akin to the second expression in Equation (A.57) to the chain of partial derivatives:

$$\frac{\partial \mathbf{x}_{l+1}}{\partial \mathbf{x}_1} = \frac{\partial \mathbf{x}_{l+1}}{\partial \mathbf{x}_l} \frac{\partial \mathbf{x}_l}{\partial \mathbf{x}_1}, \tag{A.58}$$

which is also a linear function, proving the induction step. Because we know that both terms of the product in Equation (A.58) are linear, the second term of the Equation (A.55) is as well. Together with the previous insight that the first term is exponential, this implies that it will outgrow the second in the limit, creating an infinitely-wide gap between them and thereby proving the lemma. □

Equipped with the results of Lemmas 5 and 6, we can finally prove Proposition 1:

Proof. We show that one scalar factor contained in the factorization of the gradient $\nabla_{\mathbf{x}} P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})$ tends to zero under the given assumptions, having the whole gradient become the zero vector in the limit. We begin by again factorizing the gradient $\nabla_{\mathbf{x}} P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})$ using the multivariate chain rule:

$$\nabla_{\mathbf{x}} P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x}) = \sum_{k'=1}^{K} \frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{k'}} \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_k \nabla_{\mathbf{x}} f_{\boldsymbol{\theta}}(\mathbf{x})_{k'}. \tag{A.59}$$

By Lemmas 1 and 2 we know that $f_{\boldsymbol{\theta}}$ is a component-wise strictly monotonic function on $\mathbb{P}(\mathbf{x}', d)$, which implies for the limit of $\alpha_d \to \infty$ that $\forall k \in [K] : \ f_{\boldsymbol{\theta}}(\mathbf{x})_k \to \pm\infty$. Then, Lemma 5 implies that the first factor of every part in the sum of Equation (A.59) will tend to zero in the limit. Lemma 6 ensures that the first factor approximates zero quicker than every component of the gradient $\nabla_{\mathbf{x}} f_{\boldsymbol{\theta}}(\mathbf{x})_{k'}$ potentially approaching infinity, causing the product to result in the zero vector. As this results in a sum over K zero vectors in the limit, this proves the lemma. □

A.10 Proof of Proposition 2

This section contains the proof of Proposition 2 in Section 4.1.3.

Proof. We start by rewriting the softmax probability for the k-th logit:

$$\bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_k = \frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_k)}{\sum_{k' \in [K]} \exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{k'})} = 1 - \frac{\sum_{k'' \in [K] \setminus \{k\}} \exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{k''})}{\sum_{k' \in [K]} \exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{k'})}. \tag{A.60}$$

By Lemmas 1 and 2, we have shown that $f_{\boldsymbol{\theta}}$ is a component-wise strictly monotonic function on $\mathbb{P}(\mathbf{x}', d)$, so we know that for all $k' \in [K]: \; f_{\boldsymbol{\theta}}(\mathbf{x})_{k'} \to \pm\infty$ as $\alpha_d \to \infty$. We now treat the two limits $\pm\infty$ in order. Because of the assumption that d-column of $\mathbf{V}$ has no duplicate entries, this implies that there must be a $k \in [K]$ s.t. $\forall k' \neq k: \; v_{kd} > v_{k'd}$. Thus, in the limit of $f_{\boldsymbol{\theta}}(\mathbf{x})_k \to \infty$, the sum in the *denominator* of the fraction including the logit of k will tend to infinity faster than the the sum in the *numerator* not including k's logit, and thus the fraction itself will tend to 0, proving this case. In the case of $f_{\boldsymbol{\theta}}(\mathbf{x})_k \to -\infty$, the *numerator* of the fraction will tend to 0 faster than the *denominator*, having the fraction approach 0 in the limit as well, proving the second case and therefore the lemma. □

A.11 Proof of Lemma 4

This section contains the proof of Lemma 4 in Section 4.1.4.

Proof.

$$\begin{aligned}
&\lim_{\alpha\to\infty} \left|\left|\nabla_{\mathbf{x}}\mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\left[P_{\boldsymbol{\theta}}(y=k \mid \mathbf{x})\right]\big|_{\mathbf{x}=\boldsymbol{\alpha}\circ\mathbf{x}'}\right|\right|_2 && \text{(A.61)}\\
&= \lim_{\alpha\to\infty} \left|\left|\mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\left[\nabla_{\mathbf{x}}P_{\boldsymbol{\theta}}(y=k \mid \mathbf{x})\right]\big|_{\mathbf{x}=\boldsymbol{\alpha}\circ\mathbf{x}'}\right|\right|_2 && \text{(A.62)}\\
&\leq \lim_{\alpha\to\infty} \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\Big[\underbrace{\left|\left|\nabla_{\mathbf{x}}P_{\boldsymbol{\theta}}(y=k \mid \mathbf{x})\big|_{\mathbf{x}=\boldsymbol{\alpha}\circ\mathbf{x}'}\right|\right|_2}_{=\ 0\ (\text{Proposition 1})}\Big] = 0. && \text{(A.63)}
\end{aligned}$$

Because the last expression is an upper bound to the original expression and the l_2 norm is lower-bounded by 0, this proves the lemma. □

A.12 Proof of Lemma 7

This section contains the proof of Lemma 7 that is part of the proof of Theorem 1 in Section 4.1.4.

Lemma 7. *(Asymptotic behavior with softmax variance) Suppose that $f_{\boldsymbol{\theta}}^{(1)}, \ldots, f_{\boldsymbol{\theta}}^{(K)}$ are ReLU networks. Let $\mathbf{x}' \in \mathbb{R}^D$, suppose $\boldsymbol{\alpha}$ is a scaling vector and that for all k, the associated PUP $\mathbb{P}^{(k)}(\mathbf{x}', d)$ has a corresponding matrix $\mathbf{V}^{(k)}$ with no zero entries. It holds that*

$$\begin{aligned}
&\lim_{\alpha_d\to\infty}\Bigg|\Bigg|\nabla_{\mathbf{x}}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\left[P_{\boldsymbol{\theta}}(y=k \mid \mathbf{x})^2\right]\\
&\qquad - \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\left[P_{\boldsymbol{\theta}}(y=k \mid \mathbf{x})\right]^2\Big|_{\mathbf{x}=\boldsymbol{\alpha}\circ\mathbf{x}'}\Bigg|\Bigg|_2 = 0. && \text{(A.64)}
\end{aligned}$$

Proof.

$$\lim_{\alpha_d \to \infty} \Big|\Big| \nabla_{\mathbf{x}} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})} \big[P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})^2 \big] - \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})} \big[P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x}) \big]^2 \Big|_{\mathbf{x} = \boldsymbol{\alpha} \circ \mathbf{x}'} \Big|\Big|_2 \tag{A.65}$$

$$= \lim_{\alpha_d \to \infty} \Big|\Big| \frac{1}{K} \sum_{k=1}^{K} \nabla_{\mathbf{x}} \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})} \big[P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x}) \big]^2 - \nabla_{\mathbf{x}} \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})} \big[P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x}) \big]^2 \Big|_{\mathbf{x} = \boldsymbol{\alpha} \circ \mathbf{x}'} \Big|\Big|_2 \tag{A.66}$$

Apply triangle inequality $||x + y|| \leq ||x|| + ||y||$ to sum over all k:

$$\leq \lim_{\alpha_d \to \infty} \frac{1}{K} \sum_{k=1}^{K} \big|\big| \nabla_{\mathbf{x}} \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})} \big[P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})^2 \big] - \nabla_{\mathbf{x}} \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})} \big[P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x}) \big]^2 \Big|_{\mathbf{x} = \boldsymbol{\alpha} \circ \mathbf{x}'} \big|\big|_2 \tag{A.67}$$

On the first term use linearity of gradients and apply chain rule, do it in the reverse order on the second term:

$$= \lim_{\alpha_d \to \infty} \frac{1}{K} \sum_{k=1}^{K} \big|\big| \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})} \big[2 P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x}) \underbrace{\nabla_{\mathbf{x}} P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x}) \big|_{\mathbf{x} = \boldsymbol{\alpha} \circ \mathbf{x}'}}_{= \mathbf{0} \text{ (Proposition 1)}} \big] - 2 \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})} \big[P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x}) \big] \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})} \big[\underbrace{\nabla_{\mathbf{x}} P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x}) \big|_{\mathbf{x} = \boldsymbol{\alpha} \circ \mathbf{x}'}}_{= \mathbf{0} \text{ (Proposition 1)}} \big] \big|\big|_2 = 0. \tag{A.68}$$

We can see that due to an intermediate result of Proposition 1, i.e. that $\nabla_{\mathbf{x}} P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})$ approaches the zero vector in the limit, the innermost gradients tend to zero, bringing the whole expression to zero. Because the final is an upper bound to the original expression and because the l_2 norm has a lower bound of 0, this proves the lemma. □

A.13 Proof of Lemma 8

This section contains the proof of Lemma 8 that is part of the proof of Theorem 1 in Section 4.1.4.

Lemma 8. *(Asymptotic behavior for predictive entropy) Suppose that $f_{\boldsymbol{\theta}}^{(1)}, \ldots, f_{\boldsymbol{\theta}}^{(K)}$ are ReLU networks. Let $\mathbf{x}' \in \mathbb{R}^D$, suppose $\boldsymbol{\alpha}$ is a scaling vector and that for all k, the associated PUP $\mathbb{P}^{(k)}(\mathbf{x}', d)$ has a corresponding matrix $\mathbf{V}^{(k)}$ with no zero entries. It holds that*

$$\lim_{\alpha_d \to \infty} \Big|\Big| \nabla_{\mathbf{x}} \mathrm{H} \Big[\mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})} \big[P_{\boldsymbol{\theta}}(y \mid \mathbf{x}) \big] \Big] \Big|_{\mathbf{x} = \boldsymbol{\alpha} \circ \mathbf{x}'} \Big|\Big|_2 = 0. \tag{A.69}$$

Proof.

$$\lim_{\alpha_d \to \infty} \left|\left|\nabla_{\mathbf{x}} \mathrm{H}\left[\mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\left[P_{\boldsymbol{\theta}}(y \mid \mathbf{x})\right]\right]\right|_{\mathbf{x}=\boldsymbol{\alpha}\circ\mathbf{x}'}\right|\right|_2 \tag{A.70}$$

$$= \lim_{\alpha_d \to \infty} \left|\left|\nabla_{\mathbf{x}} \left(\sum_{k=1}^{K} \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\left[P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})\right]\right.\right.\right.$$

$$\left.\left.\left. \cdot \log \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\left[P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})\right]\right)\right|_{\mathbf{x}=\boldsymbol{\alpha}\circ\mathbf{x}'}\right|\right|_2 \tag{A.71}$$

$$= \lim_{\alpha_d \to \infty} \left|\left| \sum_{k=1}^{K} \nabla_{\mathbf{x}} \left(\mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\left[P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})\right]\right.\right.\right.$$

$$\left.\left.\left. \cdot \log \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\left[P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})\right]\right)\right|_{\mathbf{x}=\boldsymbol{\alpha}\circ\mathbf{x}'}\right|\right|_2 \tag{A.72}$$

$$= \lim_{\alpha_d \to \infty} \left|\left| \sum_{k=1}^{K} \nabla_{\mathbf{x}} \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\left[p_{\boldsymbol{\theta}}(y = c| \mathbf{x})\right]\right.\right.$$

$$\left.\left. + \nabla_{\mathbf{x}} \left(\mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\left[P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})\right]\right) \log \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\left[P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})\right]\right|_{\mathbf{x}=\boldsymbol{\alpha}\circ\mathbf{x}'}\right|\right|_2 \tag{A.73}$$

$$= \lim_{\alpha_d \to \infty} \left|\left| \sum_{k=1}^{K} \nabla_{\mathbf{x}} \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\left[p_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})\right]\right.\right.$$

$$\left.\left. \cdot \left(1 + \log \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\left[P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})\right]\right)\right|_{\mathbf{x}=\boldsymbol{\alpha}\circ\mathbf{x}'}\right|\right|_2 \tag{A.74}$$

Apply triangle inequality to sum over all k:

$$\leq \lim_{\alpha_d \to \infty} \sum_{k=1}^{K} \left|\left|\nabla_{\mathbf{x}} \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\left[p_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})\right]\right.\right.$$

$$\left.\left. \cdot \left(1 + \log \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\left[P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})\right]\right)\right|_{\mathbf{x}=\boldsymbol{\alpha}\circ\mathbf{x}'}\right|\right|_2 \tag{A.75}$$

$$= \lim_{\alpha_d \to \infty} \sum_{k=1}^{K} \left(1 + \log \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\left[p_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})\right]\right)$$

$$\cdot \underbrace{\left|\left|\nabla_{\mathbf{x}} \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\left[p_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})\right]\right|_{\mathbf{x}=\boldsymbol{\alpha}\circ\mathbf{x}'}\right|\right|_2}_{=\ 0\ (\text{Lemma } 4)} = 0. \tag{A.76}$$

As the final result is an upper bound to the original expression and is lower-bounded by 0 due to the l_2 norm, this proves the lemma. □

A.14 Proof of Lemma 9

This section contains the proof of Lemma 9 that is part of the proof of Theorem 1 in Section 4.1.4.

Lemma 9. *(Asymptotic behavior for approximate mutual information) Suppose that* $f_{\boldsymbol{\theta}}^{(1)}, \dots, f_{\boldsymbol{\theta}}^{(K)}$ *are ReLU networks. Let* $\mathbf{x}' \in \mathbb{R}^D$*, suppose* $\boldsymbol{\alpha}$ *is a scaling vector and that for all* k*, the associated PUP* $\mathbb{P}^{(k)}(\mathbf{x}', d)$ *has a corresponding matrix* $\mathbf{V}^{(k)}$ *with no zero entries. It holds that*

$$\lim_{\alpha_d \to \infty} \left|\left| \nabla_{\mathbf{x}} \Big(\mathrm{H}\Big[\mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\big[P_{\boldsymbol{\theta}}(y \mid \mathbf{x})\big]\Big] - \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\Big[\mathrm{H}\big[P_{\boldsymbol{\theta}}(y \mid \mathbf{x})\big]\Big] \Big) \Big|_{\mathbf{x}=\boldsymbol{\alpha} \circ \mathbf{x}'} \right|\right|_2 = 0. \tag{A.77}$$

Proof.

$$\lim_{\alpha_d \to \infty} \left|\left| \nabla_{\mathbf{x}} \Big(\mathrm{H}\Big[\mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\big[P_{\boldsymbol{\theta}}(y \mid \mathbf{x})\big]\Big] - \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\Big[\mathrm{H}\big[P_{\boldsymbol{\theta}}(y \mid \mathbf{x})\big]\Big] \Big) \Big|_{\mathbf{x}=\boldsymbol{\alpha} \circ \mathbf{x}'} \right|\right|_2 \tag{A.78}$$

$$= \lim_{\alpha_d \to \infty} \left|\left| \Big(\nabla_{\mathbf{x}} \mathrm{H}\Big[\mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\big[P_{\boldsymbol{\theta}}(y \mid \mathbf{x})\big]\Big] - \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\Big[\nabla_{\mathbf{x}}\mathrm{H}\big[P_{\boldsymbol{\theta}}(y \mid \mathbf{x})\big]\Big] \Big) \Big|_{\mathbf{x}=\boldsymbol{\alpha} \circ \mathbf{x}'} \right|\right|_2 \tag{A.79}$$

Applying chain rule and intermediate result of Proposition 1:

$$= \lim_{\alpha_d \to \infty} \left|\left| \nabla_{\mathbf{x}} \mathrm{H}\Big[\mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\big[P_{\boldsymbol{\theta}}(y \mid \mathbf{x})\big]\Big] - \mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\Big[\sum_{k=1}^{K} \big(1 + \log P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})\big) \underbrace{\nabla_{\mathbf{x}} P_{\boldsymbol{\theta}}(y = k \mid \mathbf{x})}_{= \mathbf{0} \text{ (Proposition 1)}}\Big] \Big|_{\mathbf{x}=\boldsymbol{\alpha} \circ \mathbf{x}'} \right|\right|_2 \tag{A.80}$$

$$= \underbrace{\lim_{\alpha_d \to \infty} \left|\left| \nabla_{\mathbf{x}} \mathrm{H}\Big[\mathbb{E}_{p(\boldsymbol{\theta}|\mathbb{D})}\big[P_{\boldsymbol{\theta}}(y \mid \mathbf{x})\big]\Big] \Big|_{\mathbf{x}=\boldsymbol{\alpha} \circ \mathbf{x}'} \right|\right|_2}_{\text{(Lemma 8)}} = 0. \tag{A.81}$$

As the final result is an upper bound to the original expression and the l_2 norm provides a lower bound of 0, this proves the lemma. □

B | Experimental Appendix

> "*Machine learning has become alchemy.*"
>
> —Ali Rahimi in his NIPS 2017 Test of Time Award Talk.

Thesis	Appendix
Section 3.2.2	Appendix B.1
Section 4.1.5	Appendix C.4.2
Section 4.2.2	Appendices B.3 and B.4
Section 4.2.6	Appendix B.5
Section 4.2.7	Appendix B.6
Section 5.4.1	Appendices B.7 and B.8
Section 6.2	Appendix B.9
Section 6.2.2	Appendix B.10

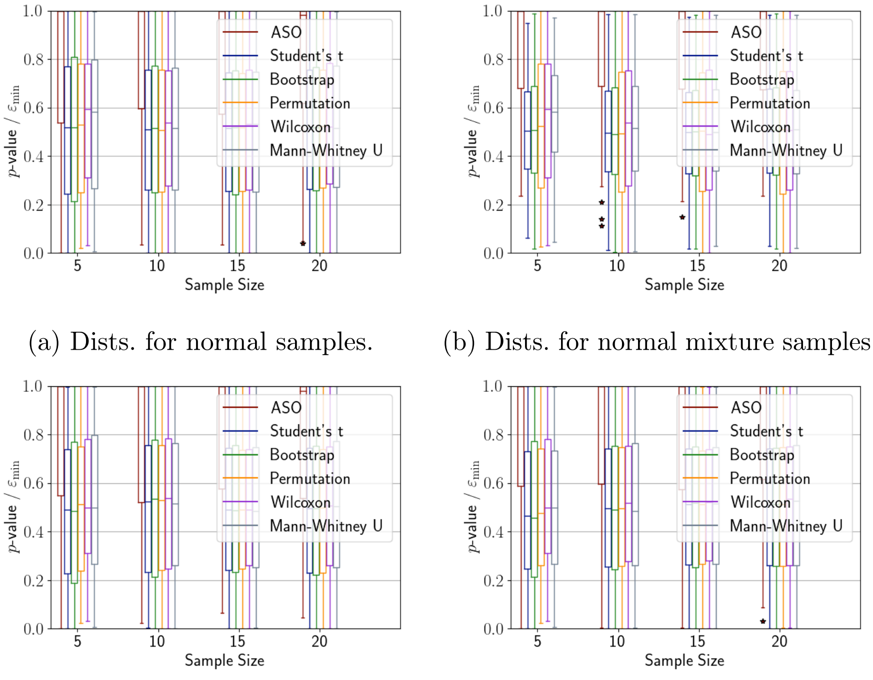
Table B.1: Correspondences between sections of the empirical appendix and thesis chapters.

This appendix involves a collection of additional empirical results stemming from the experiments in the different chapters. An overview over the contents and their correspondence to thesis chapters is given in Table B.1. For more details regarding the reproducibility of experiments (hyperparameters, experimental settings etc.) refer to Appendix C.

B.1 Additional Error Rate Experiments

We use this section to further shed light on the results in Figure 3.5.

Test Score Distributions. Instead of showing the Type I error rates based on thresholded test results, we instead plot the distributions over test scores in Figure B.1. We can observe that the lower ends of the interquartile range of ε_{min} distributions are either the same or higher than the ones for p-values (they do not need to be centered around 0.5 since ε_{min} is an upper bound to ε_{W_2}), explaining the lower Type I error rate.



(a) Dists. for normal samples.

(b) Dists. for normal mixture samples.

(c) Dists. for Laplace samples.

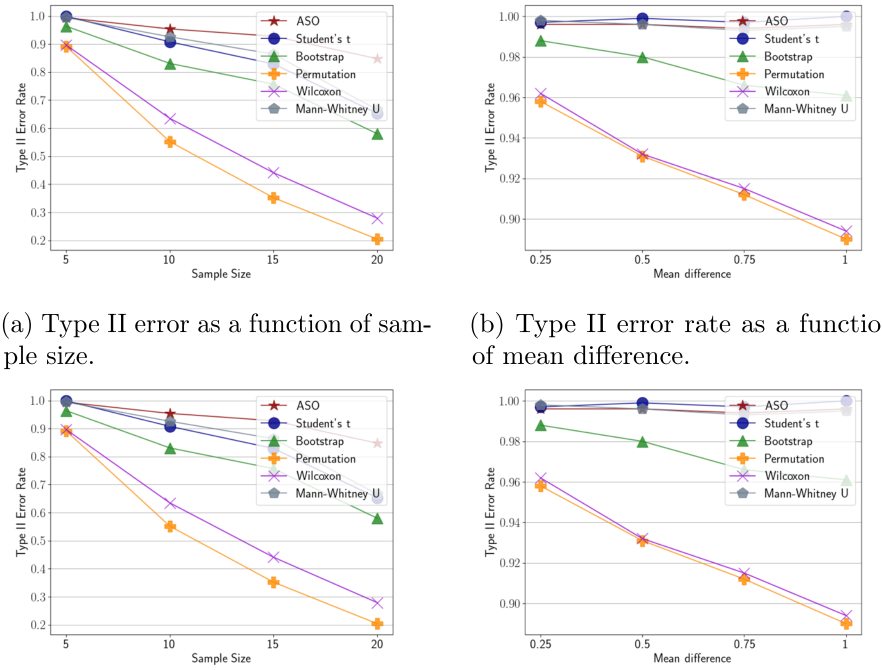
(d) Dists. for Rayleigh samples.

Figure B.1: Comparing test score distributions for different tests and distributions as a function of sample size.

Type II Error Rate Experiments. We furthermore test the Type II error rates on samples from different distributions in Figure B.2, sampling the score samples 500 times for ASO and 1000 times for the other tests from $\mathcal{N}(0.5, 1.5^2)$ and $\mathcal{N}(0, 1.5^2)$,[88] respectively, for a p-value threshold of 0.05 and $\varepsilon_{\min}$ threshold of 0.2. We see that the Type II error rate decreases with increasing sample size (Figures B.2a and B.2c), but is less sensitive for increasing mean difference than other tests (Figures B.2b and B.2d). Generally, we can observe the behavior to be very similar to Student's-t and Mann-Whitney U test.

Error Rates by Rejection Threshold. Lastly, we report the Type I and II error rates on the tested distributions using different Type I / II error rates. In Tables B.2, B.5, B.8 and B.9, we see that ASO achieves lower error rates than other tests in almost all scenarios when faced with the fame threshold. Naturally, these thresholds cannot be interpreted the same for ASO and the other significance tests. Nevertheless, we can see that a threshold of $\tau = 0.2$ seems to roughly correspond to a p-value threshold of 0.05 in terms of Type I error rate. Type II error rates are given in

[88] For the normal mixture, only the second mixture component is varied.



(a) Type II error as a function of sample size.

(b) Type II error rate as a function of mean difference.

(c) Type II error as a function of sample size.

(d) Type II error rate as a function of mean difference.

Figure B.2: Measuring the Type II error rate of the considered tests on normal and normal mixture distributions as a function of sample size Figures B.2a and B.2c and mean differences Figures B.2b and B.2d.

Tables B.3, B.4, B.6 and B.7. Here the difference between ASO and the other tests is not quite as pronounced, however, it always incurs higher error rates.

B.2 Synthetic Data Experiments

This sections provides more details on the results in Section 4.1.5. All of the plots produced can be found in Figures B.3 and B.4, where uncertainty values where plotted for different ranges depending on the metric (variance: 0-0.25; (negative) entropy: 0-1; mutual information: $4-5$; $(1-)$ max. prob: $0-0.5$), with deep purple signifying high uncertainty and white signifying low uncertainty / high certainty. We can see in Figure B.3 that maximum softmax probability and predictive entropy behave quite similarly, forming a tube-like region of high uncertainty along what appear to be the decision boundary. In both cases, the region appears to be sharper in the case of maximum softmax probability (right column) and also more defined after additional temperature scaling (bottom row). For all models and metrics, we see that the gradient magnitude decreases and approaches zero away from the training data

Sample Size	τ	ASO	Student's t	Bootstrap	Permutation	Wilcoxon	Mann-Whitney U
5	.05	**.020**	.048	.085	.029	.029	.056
	.10	**.034**	.093	.149	.079	.088	.085
	.20	**.006**	.212	.241	.197	.160	.159
	.30	**.094**	.299	.322	.286	.236	.284
	.40	**.146**	.396	.403	.370	.315	.348
	.50	**.216**	.483	.483	.468	.490	.498
10	.05	**.004**	.055	.077	.058	.051	.048
	.10	.014	.103	.130	**.110**	.113	**.100**
	.20	**.038**	.196	.215	.201	.192	.194
	.30	**.084**	.282	.300	.285	.261	.272
	.40	**.138**	.394	.398	.395	.387	.378
	.50	**.204**	.409	.486	.491	.499	.479
15	.05	**.002**	.059	.072	.057	.051	.052
	.10	**.014**	.106	.123	.104	.095	.113
	.20	**.042**	.198	.215	.199	.186	.196
	.30	**.080**	.303	.309	.303	.295	.304
	.40	**.136**	.395	.400	.392	.371	.368
	.50	**.190**	.482	.485	.479	.470	.468
20	.05	**.004**	.046	.058	.047	.043	.047
	.10	**.006**	.095	.105	.093	.085	.092
	.20	**.028**	.181	.196	.177	.171	.183
	.30	**.074**	.280	.290	.289	.284	.273
	.40	**.120**	.384	.389	.381	.372	.394
	.50	**.170**	.479	.478	.473	.477	.481

Table B.2: Type I error rates for samples drawn from a normal distribution as a function of sample size and different rejection thresholds.

(yellow / green plots), except for the cases discussed in Section 4.1.5.

In the next figure, Figure B.4, we observe the uncertainty surfaces for models using multiple network instances. For the remaining models it is interesting to see that class variance (left column) didn't seem to produce significantly different values across the feature space except for the anchored ensemble. For predictive entropy (central column), we can see a similar behavior compared to the single-instances models. Interestingly, the "fuzziness" of the high-uncertainty region increases with the ensemble and becomes increasing large with its anchored variant. Nevertheless, regions with static levels of certainty still exist in this case. For the mutual information plots (right column), epistemic uncertainty is lowest around the training data, where the model is best specified, which creates another tube-like region of high confidence even where there is no training data, an effect that is reduced with the neural ensemble and almost completely solved by the anchored ensemble. For all metrics, we see a magnitude close to zero for the uncertainty gradient away from the training data, except for the decision boundaries, as discussed in Section 4.1.5.

Sample Size	τ	ASO	Student's t	Bootstrap	Permutation	Wilcoxon	Mann-Whitney U
5	.05	.942	.883	**.796**	.918	.925	.875
	.10	.916	.786	**.714**	.802	.792	.819
	.20	.870	.623	**.585**	.649	.691	.694
	.30	.792	.512	**.480**	.521	.597	.539
	.40	.714	.399	**.309**	.421	.498	.470
	.50	.650	**.302**	.315	.318	.387	.391
10	.05	.978	.836	**.791**	.853	.864	.840
	.10	.950	.703	**.695**	.737	.743	.741
	.20	.868	.580	**.551**	.58	.595	.576
	.30	.802	.428	**.41**	.429	.462	.453
	.40	.708	.330	**.328**	.327	.347	.329
	.50	.604	**.223**	.223	.229	.272	.251
15	.05	.984	.769	.734	.781	.788	.787
	.10	.905	.643	**.615**	.646	.672	.639
	.20	.840	**.470**	.455	.480	.493	.481
	.30	.716	.348	**.340**	.350	.355	.365
	.40	.610	**.244**	.245	.246	.276	.261
	.50	.486	.177	.176	**.175**	.185	.192
20	.05	.976	.732	**.709**	.736	.750	.747
	.10	.946	.601	**.586**	.601	.614	.610
	.20	.848	.406	**.396**	.410	.421	.410
	.30	.704	.277	**.268**	.272	.299	.289
	.40	.508	**.200**	**.201**	.198	.221	.206
	.50	.444	**.144**	**.144**	.147	.156	.152

Table B.3: Type II error rates for normal samples as a function of sample size and different rejection thresholds.

B.3 Sub-Sampling of Training Sets

Since we sub-sample some of the data splits in Table 4.1, this bears the dangers of producing unnatural samples of text. For that reason, we use this appendix to describe the sampling strategies used for the methodology in Section 4.2.2 in more detail.

Sub-Sampling Procedure. The procedure for sub-sampling text is that sequences are first placed into buckets of the same label, then into sub-buckets of the same length. Then, the sampling procedure consists of first drawing a label based on the observed label frequencies, after which the draw of sequence length, proportional to the frequency of this length inside the bucket, determines the final bucket from which a sequence is again drawn uniformly. Lastly, the process for token classification involves the grouping into sequences by length at the highest level. Inside a bucket, a sequence is not drawn uniformly, but with a probability according to the *alignment* of the sequence's labels with the overall corpus label distribution. This alignment is calculated for each sequence by evaluating the expected log-probability of the sequence's la-

Difference	τ	ASO	Student's t	Bootstrap	Permutation	Wilcoxon	Mann-Whitney U
.25	.05	.984	.925	**.857**	.941	.945	.930
	.10	.954	.846	**.781**	.859	.844	.881
	.20	.914	.705	**.659**	.721	.761	.768
	.30	.872	**.585**	.554	.606	.680	.622
	.40	.800	.482	**.462**	.489	.594	.548
	.50	.714	**.381**	.387	.394	.480	.465
.50	.05	.966	.888	**.805**	.918	.920	.883
	.10	.932	.784	**.700**	.811	.794	.830
	.20	.870	.616	**.570**	.652	.696	.698
	.30	.812	.500	**.477**	.523	.602	.535
	.40	.722	.406	**.397**	.426	.505	.466
	.50	.606	**.313**	.315	.326	.411	.401
.75	.05	.934	.822	**.707**	.883	.885	.822
	.10	.896	.699	**.610**	.725	.710	.764
	.20	.798	.514	**.469**	.561	.599	.607
	.30	.702	.407	**.370**	.421	.515	.455
	.40	.590	.308	**.300**	.325	.406	.375
	.50	.482	**.223**	**.222**	.237	.303	.295
1.00	.05	.870	.739	**.609**	.850	.850	.743
	.10	.796	.585	**.488**	.678	.655	.659
	.20	.712	.386	**.327**	.449	.497	.487
	.30	.580	.257	**.232**	.289	.388	.307
	.40	.504	.178	**.170**	.194	.278	.229
	.50	.384	**.115**	**.115**	.128	.189	.176

Table B.4: Type II error rates for normal samples as a function of mean difference and different rejection thresholds.

bel distribution w.r.t. to the label distribution of the corpus (i.e., the cross-entropy). The scores for all same-length sequences in a bucket are then normalized into a $[0, 1]$ interval in order to enable sampling, which is similar to the two-stage procedure used in the sequence classification case.

Validation of Sub-Sampled Training Sets. We take multiple steps to validate the representativeness of our sub-sampled data splits. First, we plot the distributions of the 50 most frequent types in the original corpus in Figure B.5, where we see that distributions converge with increasing sample size. Secondly, we plot sentence length distributions in Figure B.6, where we also see increasing alignment with sample size. We plot the class distributions in Figure B.7. Lastly, we train an interpolated trigram Kneser-Ney language model (Jelinek, 1980; Ney et al., 1994) with uniform interpolation weights trained on the original training set using the SRILM tool (Stolcke, 2002) and sub-word tokens produced by the corresponding Bert tokenizer, sub-sample multiple splits and compare their perplexity scores to those of the original corpus in Table B.10. While n-gram perplexities of sub-sampled training

Sample Size	τ	ASO	Student's t	Bootstrap	Permutation	Wilcoxon	Mann-Whitney U
5	.05	**.000**	**.000**	.012	.028	.026	.003
	.10	**.000**	.013	.035	.079	.085	.004
	.20	**.000**	.069	.104	.179	.153	.049
	.30	**.008**	.169	.213	.281	.208	.160
	.40	**.024**	.338	.358	.363	.305	.244
	.50	**.058**	.494	.493	.483	.484	.478
10	.05	**.000**	.007	.018	.059	.049	.011
	.10	**.000**	.031	.050	.110	.109	.030
	.20	**.004**	.102	.121	.205	.188	.109
	.30	**.008**	.221	.229	.302	.273	.211
	.40	**.034**	.347	.349	.398	.379	.351
	.50	**.070**	.511	.515	.506	.491	.495
15	.05	**.000**	.006	.007	.055	.048	.004
	.10	**.000**	.022	.033	.106	.097	.017
	.20	**.002**	.103	.118	.194	.202	.095
	.30	**.006**	.215	.220	.301	.308	.208
	.40	**.028**	.356	.366	.415	.404	.328
	.50	**.082**	.501	.499	.496	.502	.501
20	.05	**.000**	.006	.007	.048	.045	.005
	.10	**.000**	.019	.027	.088	.085	.021
	.20	**.000**	.104	.109	.200	.187	.097
	.30	**.006**	.214	.218	.307	.289	.221
	.40	**.032**	.363	.369	.412	.390	.349
	.50	**.082**	.494	.495	.492	.496	.485

Table B.5: Type I error rates for normal mixture samples as a function of sample size and different rejection thresholds.

sets do lie over the ones of the original data, they are still upper-bounded by the in-distribution test set perplexities. Furthermore, this verification was not aimed to give the most precise results, as also the scoring using an n-gram model can be rather crude. Thus, with all these results, we conclude that our sub-sampling procedure produces sufficiently representative samples of the original data for the different tasks discussed.

B.4 Selection of OOD Test Sets

In this appendix section, we present additional evidence that the OOD test splits shown in Table 4.1 are sufficiently different from the training data—meaning, out-of-distribution—to enable our chosen methodology. To that end, we re-use similar ideas as described in Appendix B.3, but with the opposite goal. In Figure B.9, we plot the distribution of sequence lengths of the training set compared with the OOD test set, with the same done for the most frequent 25 types in Figure B.10 and class labels in Figure B.8. Lastly, we again use a interpolated Kneser-Ney trigram language model to compute the perplexity of the training compared to the OOD test

Sample Size	τ	ASO	Student's t	Bootstrap	Permutation	Wilcoxon	Mann-Whitney U
5	.05	1.000	.999	.964	**.892**	.897	.995
	.10	1.000	.962	.874	.728	**.697**	.985
	.20	.994	.747	.640	**.474**	.525	.870
	.30	.976	.476	.422	**.299**	.426	.579
	.40	.896	.252	.234	**.206**	.326	.414
	.50	.748	**.117**	**.118**	.122	.222	.280
10	.05	1.000	.908	.831	**.552**	.635	.926
	.10	.996	.721	.641	**.354**	.419	.730
	.20	.954	.390	.354	**.186**	.247	.407
	.30	.828	.191	.180	**.108**	.156	.219
	.40	.642	.089	.087	**.068**	.089	.107
	.50	.452	.034	**.031**	.037	.056	.052
15	.05	.996	.829	.757	**.352**	.441	.864
	.10	.990	.568	.517	**.213**	.272	.628
	.20	.928	.251	.234	**.087**	.129	.298
	.30	.774	.099	.091	**.033**	.058	.116
	.40	.498	.027	.026	**.019**	.034	.044
	.50	.276	**.009**	**.010**	**.010**	.013	.014
20	.05	1.000	.653	.580	**.204**	.279	.666
	.10	.980	.359	.333	**.105**	.162	.392
	.20	.848	.107	.101	**.035**	.064	.147
	.30	.586	.038	.035	**.013**	.022	.047
	.40	.344	.010	.010	**.008**	.013	.017
	.50	.130	**.003**	**.003**	**.004**	.006	.006

Table B.6: Type II error rates for normal mixture samples as a function of sample size and different rejection thresholds.

set in Table B.10. In all cases, OOD n-gram perplexities lie much over the training or sub-sampled data perplexities. Except for Finnish, they are also widely different from the test set perplexities. In that exceptional cases, an explanation could be given by the highly agglutinative nature of Finnish, increasing the sparsity of the language despite the subword tokenization.

B.5 Additional Scatter Plots

This section provides some additional scatter plots for the experiments in Section 4.2.6. For all plots presented here as well as Figure 4.5, some slight jitter sampled from $\mathcal{N}(0, 0.01)$ was added to x and y-coordinates to increase readability of overlapping points.

Clinc Plus. In Figures B.11a and B.12a, we can see that the variational Bert model actually *degrades* in performance as the more training data is added, both on a task and uncertainty dimensions, while other models stay relatively constant. The same trend can be detected using the sequence-level Kendall's τ for Clinc Plus. We suspect that the smallest training size of $10k$ examples does already

Diff.	τ	ASO	Student's t	Bootstrap	Permutation	Wilcoxon	Mann-Whitney U
.25	.05	1.000	.997	.988	.958	.962	.998
	.10	.998	.988	.960	.894	**.882**	.994
	.20	.996	.903	.856	**.754**	.792	.945
	.30	.978	.762	.727	**.643**	.724	.814
	.40	.940	.594	.576	**.530**	.621	.704
	.50	.886	**.424**	**.424**	.444	.532	.563
.50	.05	.998	.999	.980	**.931**	.932	.996
	.10	.998	.978	.931	.820	**.802**	.990
	.20	.996	.849	.775	**.647**	.695	.905
	.30	.976	.659	.603	**.511**	.611	.724
	.40	.928	.458	.438	**.407**	.504	.577
	.50	.840	**.284**	.287	.310	.395	.449
.75	.05	1.000	.997	.966	**.912**	.915	.993
	.10	.998	.966	.901	.769	**.746**	.985
	.20	.994	.802	.707	**.553**	.623	.886
	.30	.974	.547	.497	**.397**	.516	.651
	.40	.922	.355	.337	**.286**	.407	.485
	.50	.824	**.191**	**.191**	.198	.305	.363
1.00	.05	1.000	1.000	.961	**.890**	.894	.995
	.10	1.000	.958	.868	.714	**.682**	.989
	.20	.996	.715	.617	**.445**	.505	.872
	.30	.962	.432	.380	**.291**	.419	.545
	.40	.870	.253	.235	**.204**	.308	.408
	.50	.702	**.120**	**.120**	.132	.208	.263

Table B.7: Type II error rates for normal mixture samples as a function of mean difference between two of the mixture components and different rejection thresholds.

provide enough data for models to converge to similar solutions even after adding more data, and that the variational Bert alone might be prone to overfitting in this case.

Dan+. Results for the Danish dataset are shown in Figures B.11b and B.12b. It is apparent that LSTM-based models stay mostly constant in their predictive performance, with the largest gains observed by the LSTM ensemble. We can also observe the DDU and variational Bert to increase both in task performance and uncertainty quality with increasing training data. Interestingly, we can see for the SNGP Bert that uncertainty estimates become more indicative of OOD with more training samples, but mostly only using predictive entropy and the maximum probability score. This might indicate that in these cases, the model actually achieves the desired distance-awareness posed by Liu et al. (2023). In Figure B.14b, we can see a similar behavior of the SNGP-Bert and its metrics w.r.t. to the sequence-level correlation. Also, we

Sample Size	τ	ASO	Student's t	Bootstrap	Permutation	Wilcoxon	Mann-Whitney U
5	.05	**.022**	.053	.110	.048	.046	.066
	.10	**.038**	.117	.164	.106	.116	.097
	.20	**.088**	.223	.261	.208	.187	.169
	.30	**.124**	.319	.343	.295	.234	.286
	.40	**.154**	.427	.445	.398	.322	.379
	.50	**.218**	.509	.510	.491	.506	.508
10	.05	**.004**	.059	.077	.060	.046	.051
	.10	**.012**	.114	.142	.111	.106	.098
	.20	**.056**	.218	.236	.216	.202	.199
	.30	**.104**	.314	.330	.318	.290	.291
	.40	**.164**	.404	.407	.398	.378	.400
	.50	**.238**	.475	.475	.473	.481	.486
15	.05	**.000**	.052	.066	.048	.048	.048
	.10	**.012**	.100	.117	.103	.100	.101
	.20	**.028**	.204	.220	.199	.199	.187
	.30	**.070**	.311	.319	.303	.296	.294
	.40	**.120**	.404	.409	.402	.378	.394
	.50	**.194**	.510	.514	.511	.504	.519
20	.05	**.004**	.044	.047	.048	.057	.052
	.10	**.010**	**.099**	.113	.104	.103	.101
	.20	**.030**	.214	.232	.215	.199	.202
	.30	**.064**	.312	.325	.308	.297	.307
	.40	**.138**	.414	.413	.415	.381	.405
	.50	**.220**	.507	.505	.501	.485	.496

Table B.8: Type I error rates for samples drawn from a Laplace distribution as a function of sample size and different rejection thresholds.

see that the other Bert models and LSTM-Ensemble actually loose in uncertainty quality as more data is added.

Finnish UD. In Figures B.11c and B.12c, we observe that the AUROC and AUPR scores of different models and metrics stay largely constant across dataset sizes, which could be explained with the larger amount of training data supplied compared to Dan+. On the token-level correlation between uncertainty and loss in Figure B.13, we see the DDU Bert profiting most from more data. On a sequence-level, as depicted in Figure B.14c, the correlation appears mostly static across training set sizes, with only small gaps between in-distribution and out-of-distribution data.

Overall, it seems that the range of dataset sizes for Dan+ show the most critical differences between models, while for the dataset sizes used for Finnish UD and Clinc Plus, enough data seems to be supplied for changes to be more miniscule. This result is particularly relevant for low-resource setting, although the dependency on the task can not be disentangled from these results.

Sample Size	τ	ASO	Student's t	Bootstrap	Permutation	Wilcoxon	Mann-Whitney U
5	.05	**.012**	.054	.107	.028	.028	.054
	.10	**.034**	.108	.147	.089	.096	.088
	.20	**.076**	.203	.235	.187	.162	.165
	.30	**.110**	.319	.342	.302	.229	.291
	.40	**.146**	.423	.435	.415	.331	.360
	.50	**.198**	.532	.539	.523	.530	.524
10	.05	**.012**	.046	.062	.043	.039	.041
	.10	**.018**	.087	.107	.093	.094	.084
	.20	**.044**	.187	.206	.180	.172	.187
	.30	**.064**	.295	.314	.297	.265	.284
	.40	**.114**	.401	.405	.399	.373	.412
	.50	**.180**	.507	.514	.505	.500	.508
15	.05	**.004**	.050	.064	.049	.050	.054
	.10	**.010**	.100	.115	.100	.103	.104
	.20	**.036**	.194	.201	.182	.187	.187
	.30	**.070**	.295	.302	.287	.294	.291
	.40	**.114**	.386	.394	.379	.371	.373
	.50	**.198**	.481	.484	.487	.472	.497
20	.05	**.002**	.054	.064	.059	.055	.052
	.10	**.004**	.115	.121	.113	.103	.113
	.20	**.030**	.195	.205	.202	.187	.204
	.30	**.058**	.281	.287	.277	.283	.291
	.40	**.130**	.377	.386	.375	.368	.384
	.50	**.190**	.489	.493	.493	.468	.469

Table B.9: Type I error rates for samples drawn from a Rayleigh distribution as a function of sample size and different rejection thresholds.

B.6 Qualitative Analysis

This section provides more examples for the qualitative analysis in Section 4.2.7.

Dan+. We show more examples of the predictive entropies on samples from the Dan+ dataset in Figure B.15, where uncertainty values where jointly normalized by subtracting the mean and dividing by the standard deviation over all models and time steps. We can make the following observations: Firstly, uncertainty seems to decrease on punctuation marks such as commas and full-stops. Secondly, uncertainty appears higher on sub-word tokens and some named entities. Thirdly, DDU Bert and the LSTM ensemble produce the highest uncertainty values, which are also two of the best performing models on the task.

Finnish UD. Here, we give more examples of the analysis on the Finnish UD dataset in Figure B.16. First of all, we see that the variational LSTM and SNGP Bert seem to produce almost constant uncertainty scores, which can be explained by their suboptimal

		Sub-sampled Train ppl.↓				
Language	Train ppl.↓	$n = 100$	$n = 500$	$n = 1000$	Test ppl.↓	OOD Test ppl.↓
English	31.54	43.97 ± 2.46	44.50 ± 0.68	44.9 ± 0.4	53.11	120.32
Danish	112.73	252.52 ± 13.25	247.09 ± 3.3	249.27 ± 3.15	418.71	524.32
Finnish	116.49	257.67 ± 10.96	257.66 ± 4.7	260.36 ± 5.36	1374.76	1284.82

Table B.10: Results of using an interpolated Kneser-Ney n-gram language model on selected datasets, including sub-sampled training splits and the OOD test set. Scores of sub-sampled training sets were obtained over five different attempts.

performance in task, as shown by their results in Table 4.2. But even for the models that perform better, such as the variational Bert and the LSTM ensemble, the decomposition of predictive entropy into aleatoric and epistemic uncertainty reveals that model uncertainty generally remains low, and is overshadowed to a larger extent by the aleatoric uncertainty. We can observe that similar to Danish, uncertainty seems to be low on punctuation marks and high on subword tokens. Furthermore, aleatoric uncertainty seems to be higher on nouns and pronouns. This could be due to the sheer number of possible nouns and pronouns that could fill such a gap in a sentence.

B.7 Additional Coverage Results

We show additional plots for the experiments in Section 5.4.1, illustrating the coverage per set size-bins in Figure B.17. We can see the counterparts for Figure 5.2 using the larger M2M100$_{(1.2\text{B})}$ model in Figures B.17a and B.17b: Instead of leveling off like for the smaller model, most prediction set sizes are either in a very small range or in a size of a few ten thousand. In Figures B.17c and B.17d, we show similar plots for the two different OPT model sizes. Since in both cases, most prediction set sizes are rather small, we zoom in on the the sizes from 1 to 100. Here, we can observe a similar behavior to the smaller M2M100$_{(400\text{m})}$, gradually leveling off. We do not show similar plots for other distance metrics as they show similar trends.

B.8 Ablating Neighborhood Size and Desired Coverage

In this section, we present experiments surrounding the two most pivotal parameters of our method in Section 5.3: The desired confidence level α, as well as the number of neighbors.

	α	% Cov.	∅ Width ↓	Scc ↑	Ecg ↓
$\text{M2M100}_{(400\text{M})}$ / de → en	.1	.9442	.31	.8702	.0011
	.2	.8767	.18	.7906	8.63×10^{-5}
	.3	.7963	.12	.0000	.0016
	.4	.7058	.09	.1393	.0082
	.5	.6081	.07	.2836	.0055
	.6	.5017	.06	.1393	.0082
	.7	.3896	.05	.0000	.0091
	.8	.2800	.05	.0000	.0090
	.9	.1762	.04	.0000	.0071
$\text{M2M100}_{(400\text{M})}$ / ja → en	.1	.7453	.15	.3080	.1511
	.2	.5579	.07	.2728	.2446
	.3	.4277	.04	.2770	.2779
	.4	.3438	.03	.1212	.2438
	.5	.2749	.03	.0455	.1883
	.6	.2175	.02	.0000	.1207
	.7	.1685	.02	.0000	.0560
	.8	.1309	.01	.0000	.0117
	.9	.0989	.02	.0000	.0099
$\text{OPT}_{(350\text{M})}$ / OpenWebText	.1	.9460	.26	.8	1.85×10^{-5}
	.2	.8937	.16	.8000	.000
	.3	.8392	.10	.5000	8.74×10^{-6}
	.4	.7782	.08	.6667	.000
	.5	.7171	.06	.0000	1.19×10^{-5}
	.6	.6559	.06	.6033	.000
	.7	.5945	.05	.000	8.21×10^{-6}
	.8	.5349	.05	.4462	.000
	.9	.4757	.05	.3580	.000

Table B.11: Results for varying values of α using different models and datasets.

	K	% Cov.	∅ Width ↓	Scc ↑	Ecg ↓
$\text{M2M100}_{(400\text{M})}$ / de → en	10	.9923	.39	.9728	.0000
	25	.9563	.37	.8877	.0011
	50	.9504	.32	.8870	.0006
	75	.9444	.32	.8641	.0014
	100	.9442	.31	.8702	.0011
	200	.9422	.31	.8125	.0016
	300	.9404	.31	.8483	.0019
	500	.9389	.31	.8214	.0023
$\text{M2M100}_{(400\text{M})}$ / ja → en	10	.8013	.17	.2995	.1606
	25	.7353	.17	.2994	.1438
	50	.7540	.17	.3023	.1603
	75	.7368	.16	.3019	.1603
	100	.7453	.15	.3072	.1529
	200	.7295	.14	.2938	.1787
	300	.7192	.13	.2948	.1788
	500	.7110	.13	.2756	.1867
$\text{OPT}_{(350\text{M})}$ / OpenWebText	10	.9438	.35	.8824	.0019
	25	.9522	.33	.8333	2.06×10^{-5}
	50	.9442	.27	.0000	1.86×10^{-5}
	75	.9477	.27	.8000	1.03×10^{-5}
	100	.9460	.26	.8000	1.86×10^{-5}
	200	.9487	.28	.8571	6.20×10^{-5}
	300	.9500	.28	.8181	1.86×10^{-5}
	500	.9508	.29	.8181	1.86×10^{-5}

Table B.12: Results for varying neighborhood sizes K using different models and datasets.

Coverage Threshold. In Table B.11, we investigate the impact of different values on α on our evaluation metrics. We show that the increase in α does indeed produce the expected decrease in coverage, however with a certain degree of overcoverage for the de → en MT and the LM task. The loss in coverage always goes hand in hand with a decrease in the average prediction set width as well, as the model can allow itself to produce tighter prediction sets at the cost of higher miscoverage. As this also produces bin in which all contained instances are uncovered, this produces zero values for the SCC, while we cannot discern clear trends for the ECG.

Neighborhood Size. In Table B.12, we vary the effect of the chosen neighborhood size (with 100 being the value we use in our main experiments). We make the following, interesting observations: Coverage on the MT task seems to decrease with an increase in the neighborhood size as prediction set widths get smaller on average, with a neighborhood size around 100 striking a balance between coverage, width, computational cost and SCC / ECG. For LM, coverage seems to be mostly constant, with prediction set width hitting an inflection point for 100 neighbors. We speculate that initially there might be a benefit to considering more neighbors to calibrate $\hat{q}$, but that considering too large neighborhoods might introduce extra noise. While we found 100 to be a solid choice for the purpose of our experiments, we leave more principled ways to determine the neighborhood size to future work.

B.9 Additional Clustering Results

Cluster 1120
How many fluid ounces are in one quarter of an imperial pint?
How many fluid ounces in one Imperial pint?
How many fluid ounces in half an Imperial pint?
Cluster 920
Which famous US outlaw shot the cashier of a savings bank in Gallatin Missouri in 1869?
What famous outlaw committed the Wild West's first train robbery on July 21, 1873 in Adair, Iowa?
On July 21, 1873, Jesse James and the James-Younger gang pulled off the first successful what of the American West, in Adair Iowa?
Cluster 984
In what country was the game of golf invented?
Which ball game was invented by Dr James Naismith in Massachusetts USA in 1891?
It is generally accepted that the game of golf originated in what country?
What's a country where most people love playing rugby?
What's a country where most people love playing golf?
Cluster 811
How many colors are there in the spectrum when white light is separated?
Which part of the eye contains about 137 million light-sensitive cells in one square inch?
Which of the retina's cells can distinguish between different wavelengths of light?
In four colour process printing, which is also known as CMYK, which are the only four colours that are used?
How many colours are in the rainbow?
In art, what are the three primary colours?
What color consists of the longest wavelengths of lights visible by the human eye?
What are the three primary colours of light?

Table B.13: Contents of some randomly sampled clusters that result from the clustering procedure for TriviaQA.

Cluster 823
Where in Europe is it located?
Is it in the European Plain?
Which region of Europe is it in?
Cluster 1176
Did she have children?
Does she have any children?
Did she have any children?
Did she have any other children?
Cluster 2244
Who won the Kentucky Derby?
as he won the Derby before?
Has he raced in the Derby before?
What were the winning horse's odds?
How many Derbys have their been?
Cluster 11
Are they densities of everything the same?
What is the densest elements at regular conditions?
What is density of a substance?
What is another symbol for density?
Who gives weight per unit volume as the definition?
Where is density the same value as it's mass concentration?
To make comparisons easier what stands in for density?
What is the relative density of something that floats?
Cluster 1081
Who was murdered?
who was killed?
Who committed this murder?
who was killed?
Who was killed?
who was killed?

Table B.14: Contents of some randomly sampled clusters that result from the clustering procedure for CoQA.

In this section we take a closer look at the results of the clustering procedure described in Section 6.1.2. In our experiments, we run HDBSCAN using a minimum cluster size of three, since preliminary experiments showed this number to produce the best trade-off between the coherence of cluster contents (as evaluated in Table 6.2) and a diversity in cluster targets. This setting yields a distribution of cluster sizes shown in Figure B.18. We can see that the majority of cluster sizes are rather small, including questions on specific topics, some of which we display in Tables B.13 and B.14. Not shown are cluster sizes over 20 since the distribution quickly levels off, as well the set of all points that could not be sorted into any cluster.

After clustering and computing the average accuracy per cluster, we obtain a distribution over calibration targets, which we show with density plots in Figure B.19. Since most clusters are of size three, we can see clear modes around 0, 0.33, 0.66 and 1 for Vicuna v1.5 in Figure B.19a. For GPT-3.5 in Figure B.19b these are however less pronounced: We see that targets are often concentrated on 0 or 1, respectively. Similar spikes like in Figure B.19a are observable for both models on CoQA in Figures B.19c and B.19d. This trend is also visible when plotting the assigned calibration targets per data point in Figure B.20: While we can spot more transitionary colors between the blue and red extremes in the manifold for Figure B.20a, the colors tend more to either of the options Figure B.20b. These mode trends continue for CoQA in Figure B.20c and Figure B.20d.

B.10 Additional Calibration Results

Additional reliability plots. We show the all available reliability diagrams for the experiments in Section 6.2.2 for Vicuna-v1.5 for TriviaQA in Figure 6.4 and CoQA in Figure B.22, as well as the corresponding plots for GPT-3.5 in Figures B.23 and B.24. Sequence likelihood *can* be well-calibrated already, but this fact depends strongly on the dataset in question. And while our version of Platt scaling can improve results, it also narrows the range of confidence values to a narrow window. Verbalized uncertainty in both of variants also is not able to produce a wide variety of responses, even though this effect is slightly less pronounced for GPT-3.5. The auxiliary model is able to predict a wider array of confidence values in all settings, with the clustering variant achieving better calibration overall.

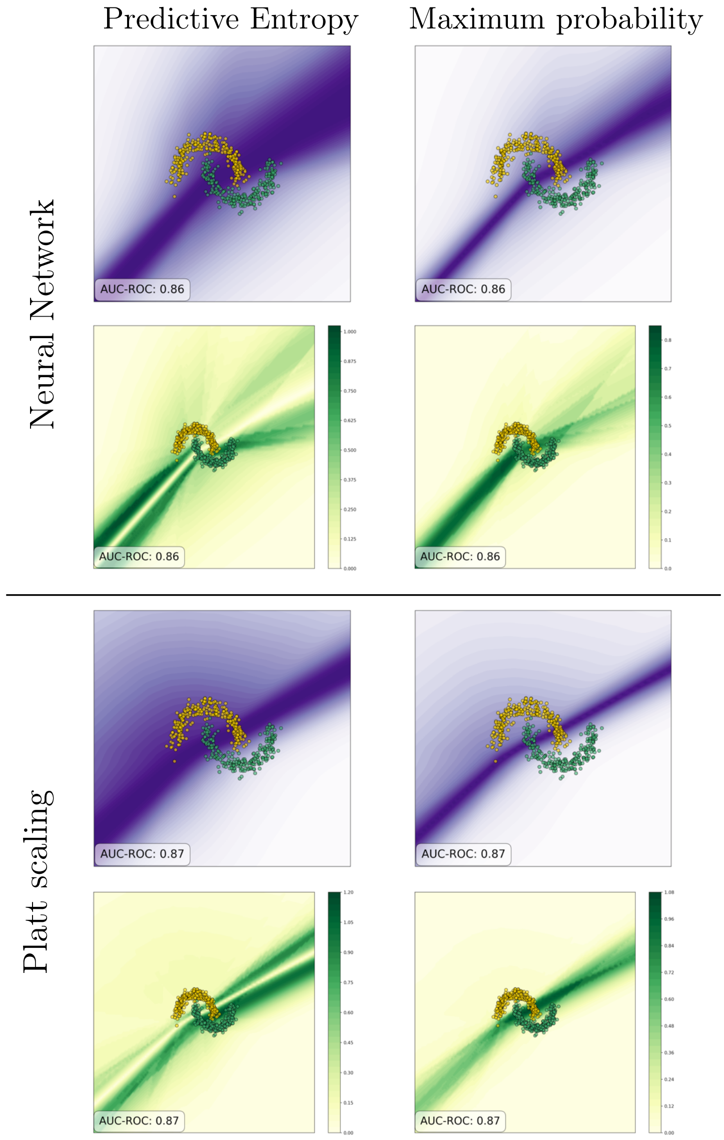


Figure B.3: Uncertainty measured by different metrics for single-instance models (purple plots) and their gradient magnitude (yellow / green plots).

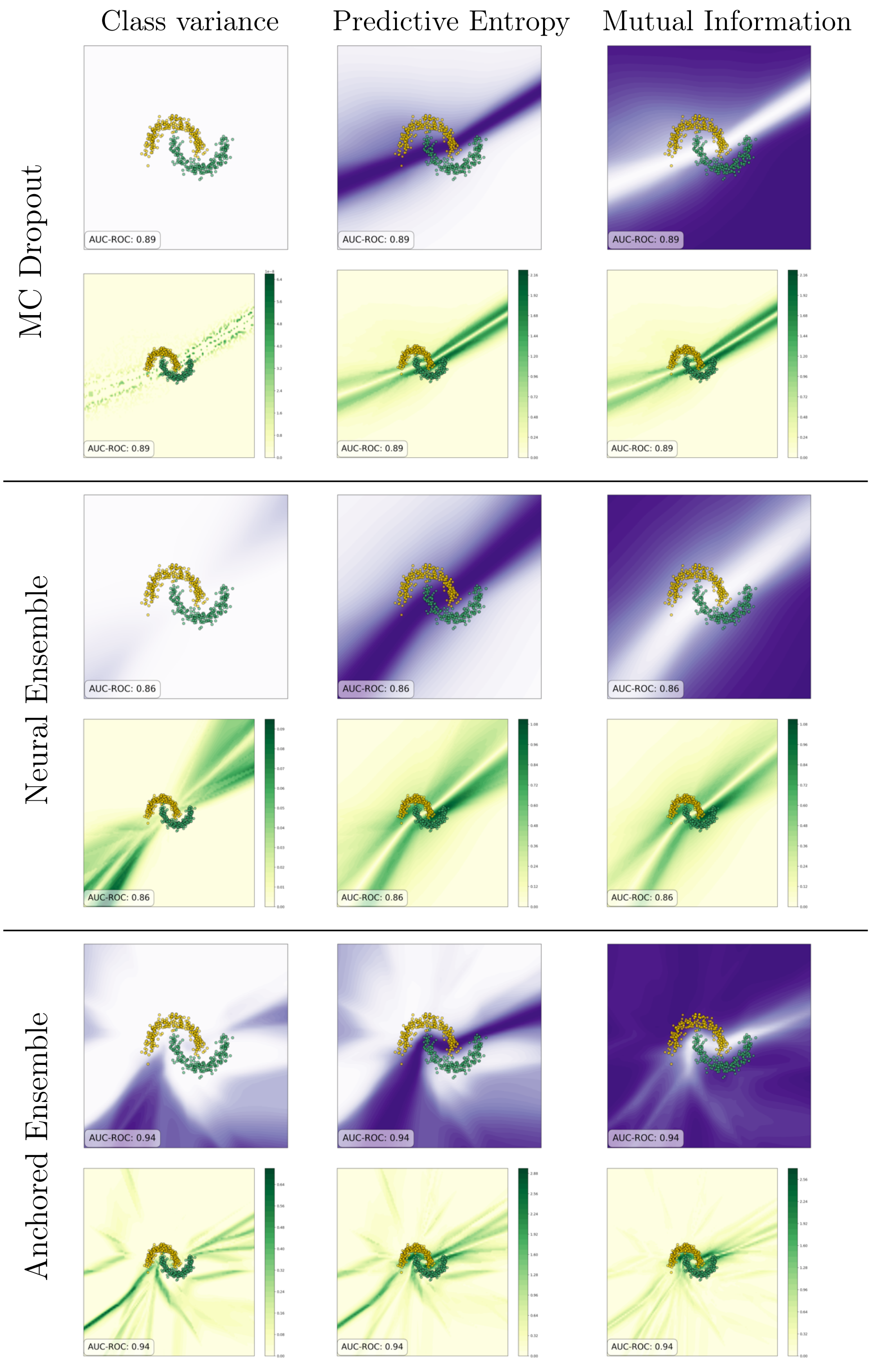


Figure B.4: Uncertainty measured by different metrics for multi-instance models (purple plots) and the gradient of the uncertainty score w.r.t. to the input (yellow / green plot).

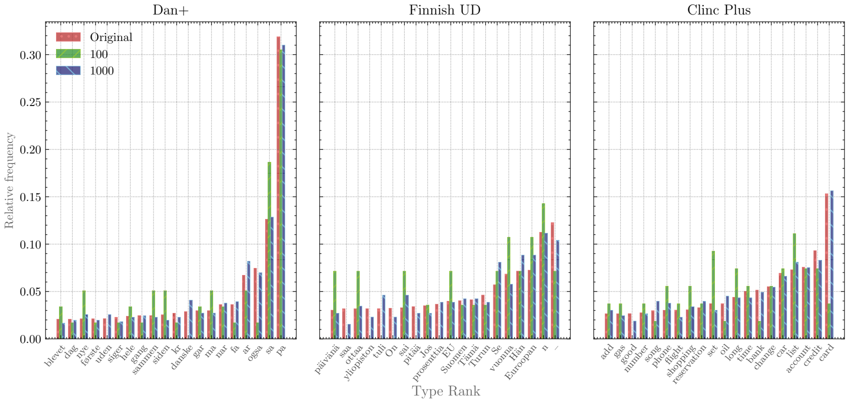


Figure B.5: Comparing the relative frequency of types in the original and sub-sampled training sets. Shown are the top 20 types in the original training set, compared to sub-sampled training sets of 100 and 1000 sequences for Dan+, Finnish UD and Clinc Plus. It is shown that while the type frequencies differ noticeably for the small dataset, already 1000 sequences suffice to approximate the original frequencies. Numbers, stopwords and the most common punctuation were removed.

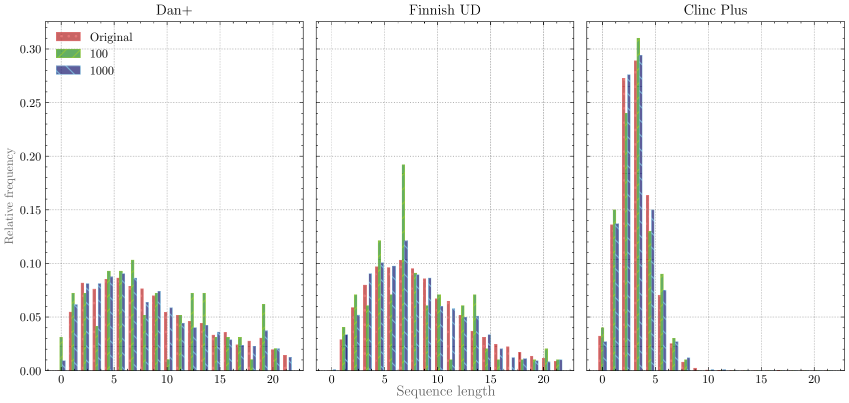


Figure B.6: Comparing the relative frequency of sequence lengths in the original and sub-sampled training sets. Shown are sequence lengths between 0 and 25 in the original test, compared to OOD test sets for Dan+, Finnish UD, Clinc Plus. Not the whole distribution is shown in all cases, with many of the OOD sentences for Dan+ being very long. For Dan+ and Finnish UD, the sentence length distributions are noticeably different. For Clinc Plus, they are very similar.

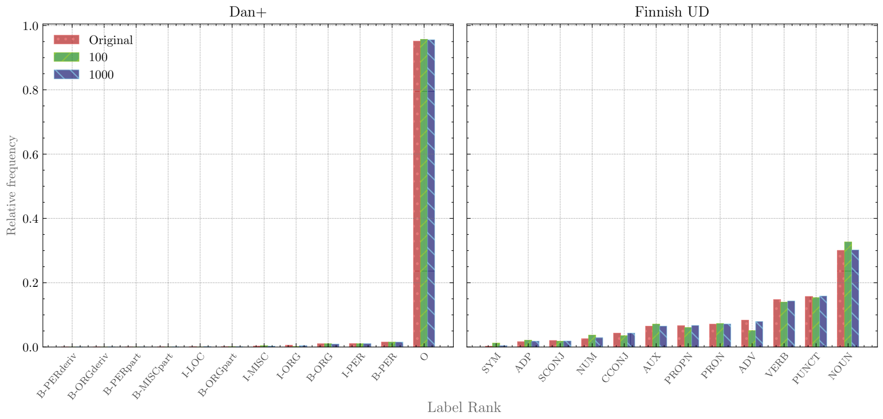


Figure B.7: Comparing the relative frequency of labels in the original training set, compared to sub-sampled training sets. Shown are frequencies for 100 and 1000 sequences. For Danish, the most frequent label by far is the neutral label indicating that no named entity is present.

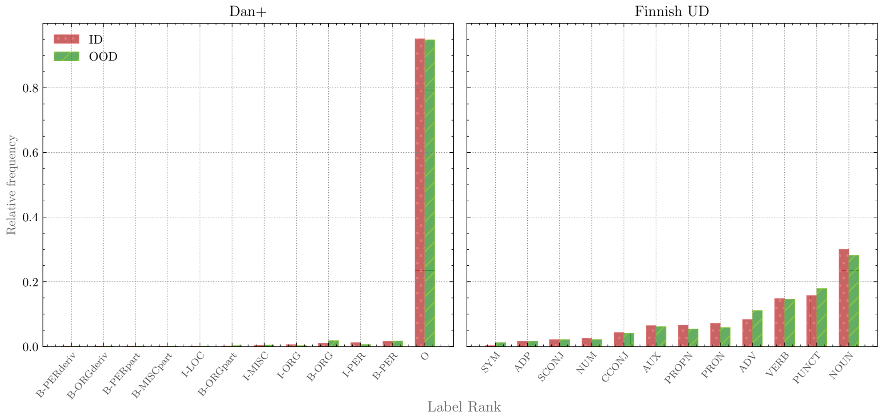


Figure B.8: Comparison of the relative class frequencies between original training set compared to the OOD test set. The proportions stay largely the same for Danish, while different more for Finnish.

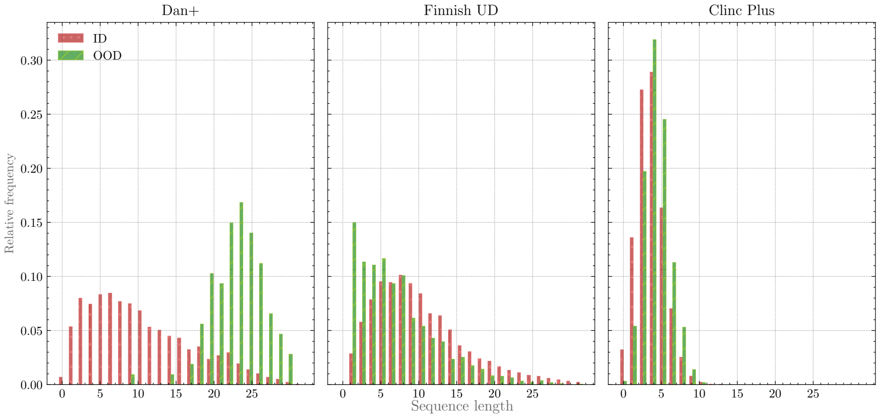


Figure B.9: Comparison of sequence length distribution between the original training set and the OOD test set. For English, the distribution of lengths of voice assistant commands is quite similar, while the differences for Dan+ and Finnish UD are more pronounced.

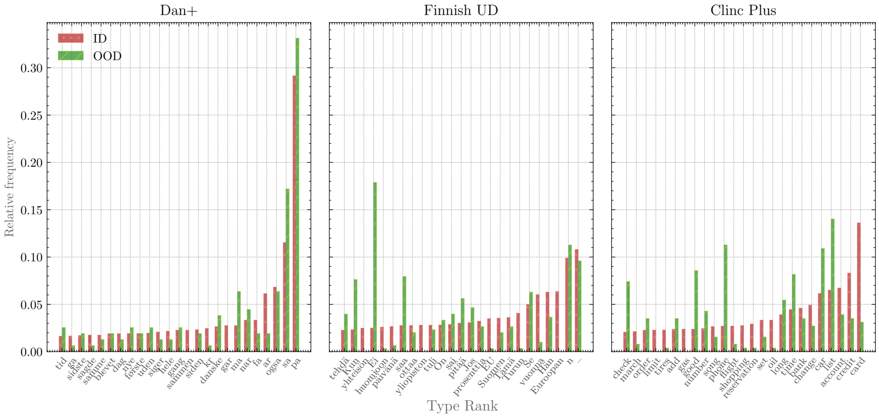
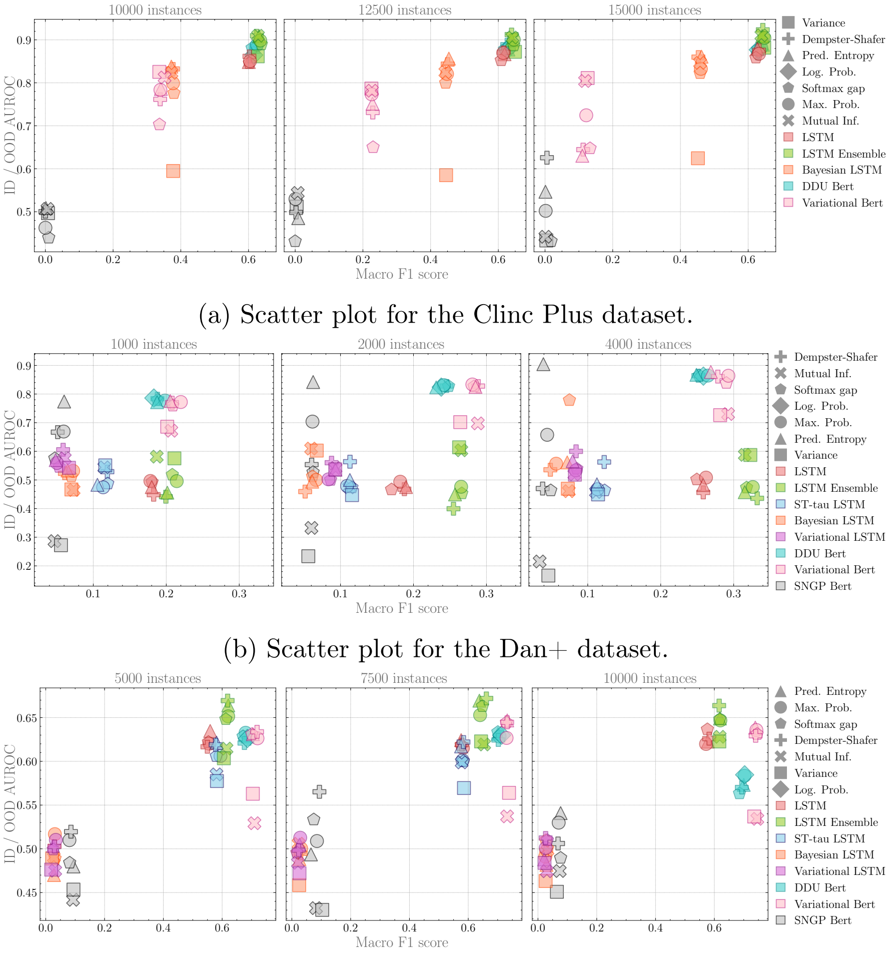


Figure B.10: Comparison of the relative frequencies of the top 25 types in the original training set compared to the OOD test set. Even among the most frequent and therefore usually common tokens, the plots show differences between the in-distribution train and out-of-distribution test set. Numbers, stopwords and the most common punctuation were removed.

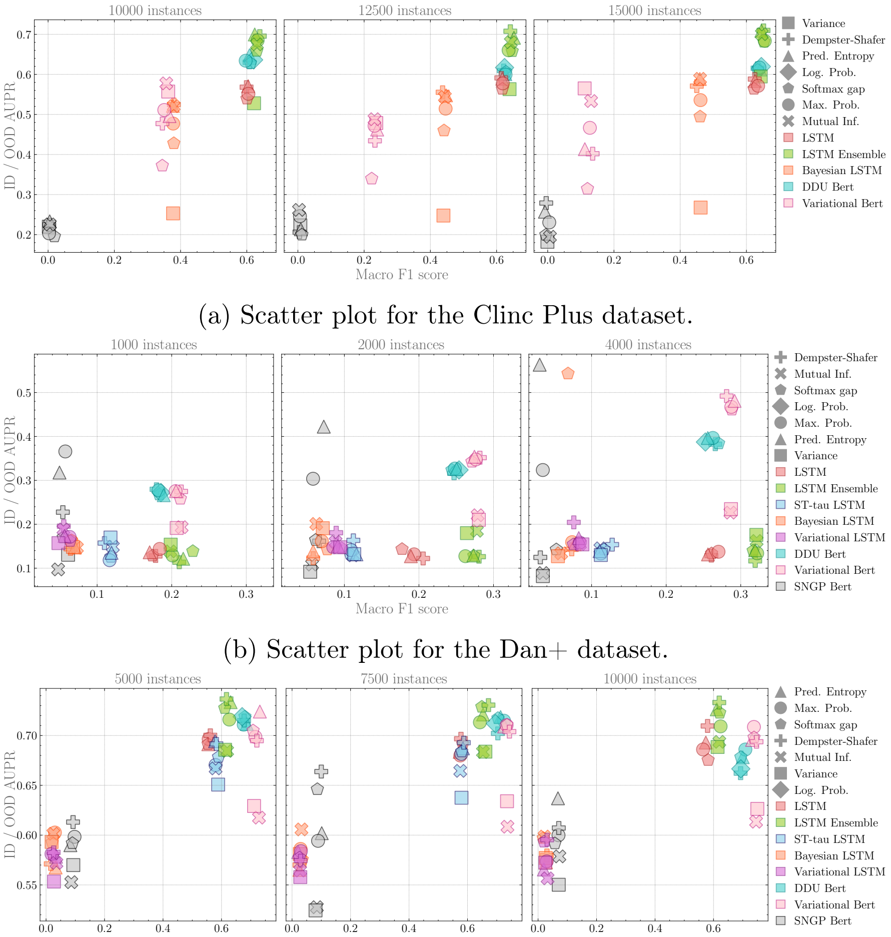


(a) Scatter plot for the Clinc Plus dataset.

(b) Scatter plot for the Dan+ dataset.

(c) Scatter plot for the Finnish UD dataset.

Figure B.11: Scatter plots showing the difference between model performance (measured by macro F_1) and the quality of uncertainty estimates using AUROC. Shown are different models and uncertainty metrics and several training set sizes on the used datasets.



(a) Scatter plot for the Clinc Plus dataset.

(b) Scatter plot for the Dan+ dataset.

(c) Scatter plot for the Finnish UD dataset.

Figure B.12: Scatters plot showing the difference between model performance (measured by macro F_1) and the quality of uncertainty estimates using AUPR. Shown are different models and uncertainty metrics and several training set sizes on the used datasets.

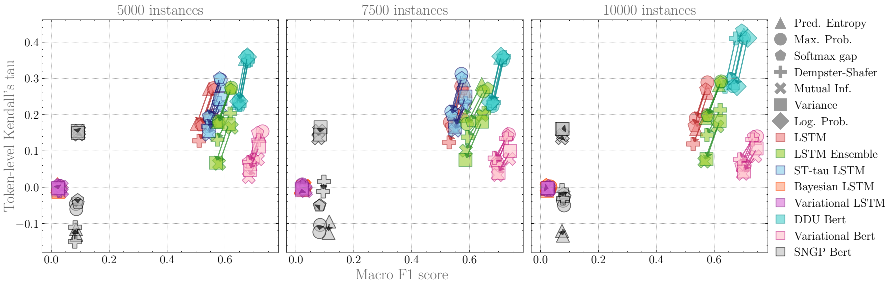
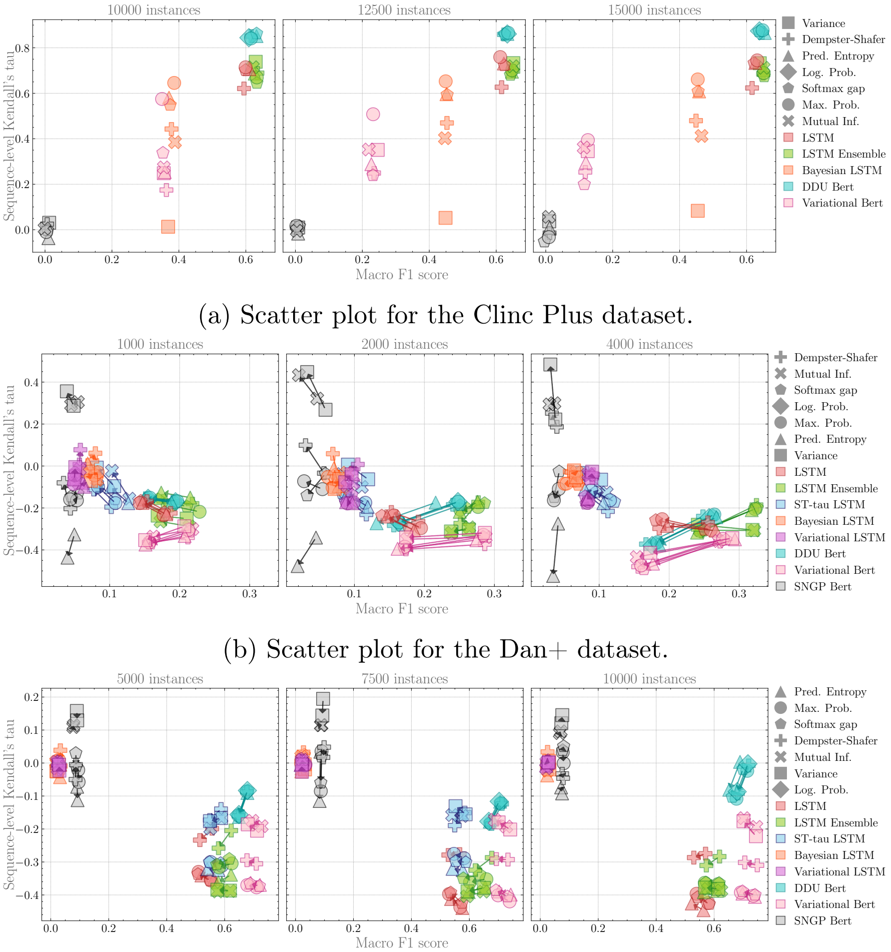


Figure B.13: Scatter plot showing the difference between model performance (measured by macro F_1) and the quality of uncertainty estimates on a token-level (measured by Kendall's τ). Results are shown for different models and uncertainty metrics and several training set sizes on the Finnish UD dataset. Arrows indicate changes between the in-distribution and out-of-distribution test set.

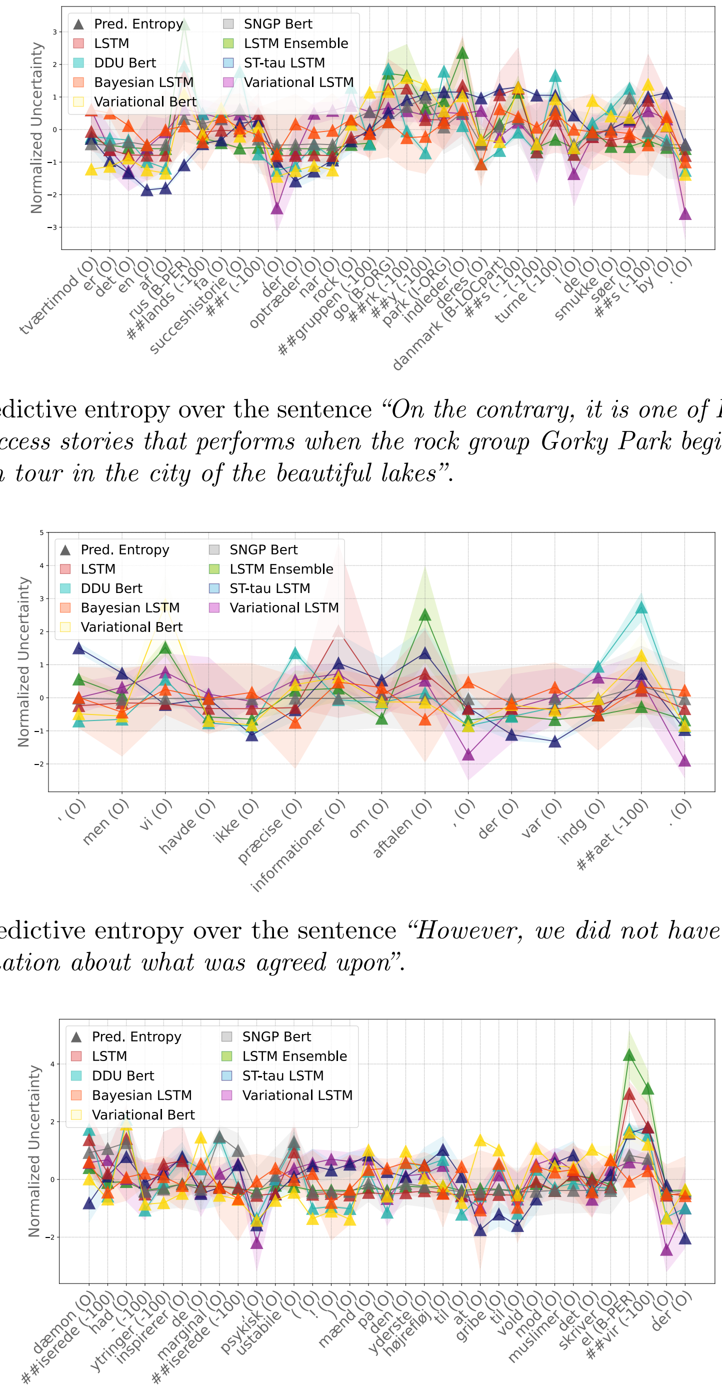


(a) Scatter plot for the Clinc Plus dataset.

(b) Scatter plot for the Dan+ dataset.

(c) Scatter plot for the Finnish UD dataset.

Figure B.14: Scatter plot showing the difference between model performance (measured by macro F_1) and the quality of uncertainty estimates on a sequence-level (measured by Kendall's τ). Results are shown for different models and uncertainty metrics and several training set sizes on the Finnish UD and Clinc Plus dataset. Arrows indicate changes between the in-distribution and out-of-distribution test set.

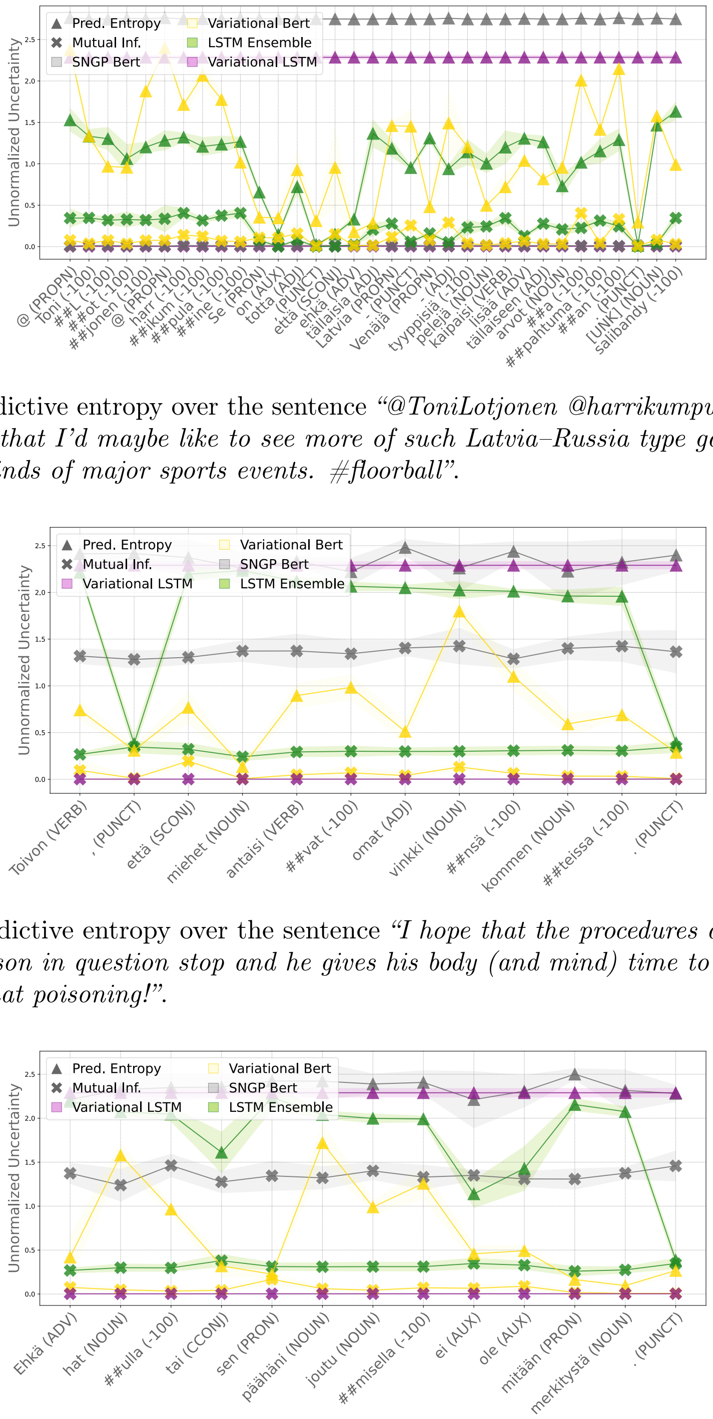


(a) Predictive entropy over the sentence *"On the contrary, it is one of Russia's few success stories that performs when the rock group Gorky Park begins their Danish tour in the city of the beautiful lakes"*.

(b) Predictive entropy over the sentence *"However, we did not have precise information about what was agreed upon"*.

(c) Predictive entropy over the sentence *"Demonizing hate speech inspires the marginalized, PSYCHOLOGY UNSTABLE (!) Men on the far right to resort to violence against Muslims. This writes Elvir, who..."*.

Figure B.15: Further examples for uncertainty estimates on single sequences. Taken from the Dan+ dataset.

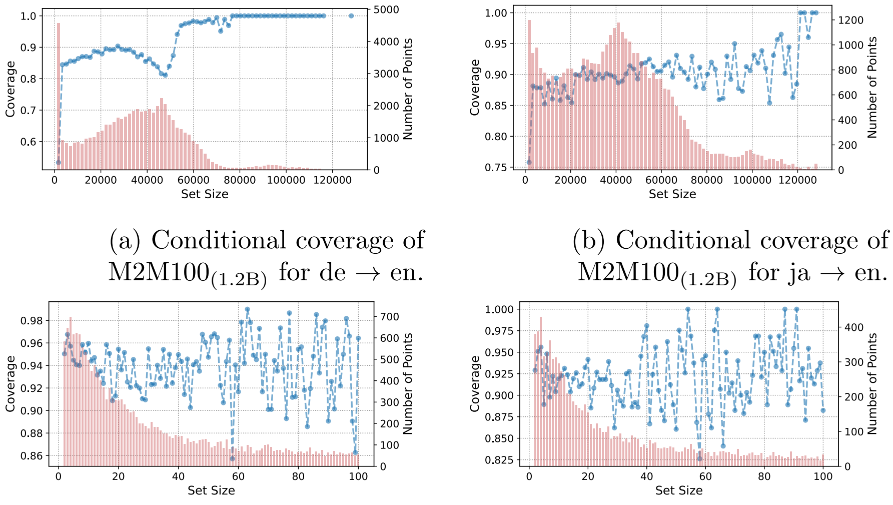


(a) Predictive entropy over the sentence *"@ToniLotjonen @harrikumpulaine It is true that I'd maybe like to see more of such Latvia–Russia type games in these kinds of major sports events. #floorball"*.

(b) Predictive entropy over the sentence *"I hope that the procedures done on the person in question stop and he gives his body (and mind) time to recover from that poisoning!"*.

(c) Predictive entropy over the sentence *"Maybe the hat or how it got on my head doesn't matter"*.

Figure B.16: Further examples for uncertainty estimates on single sequences. Taken from the Finnish UD dataset.



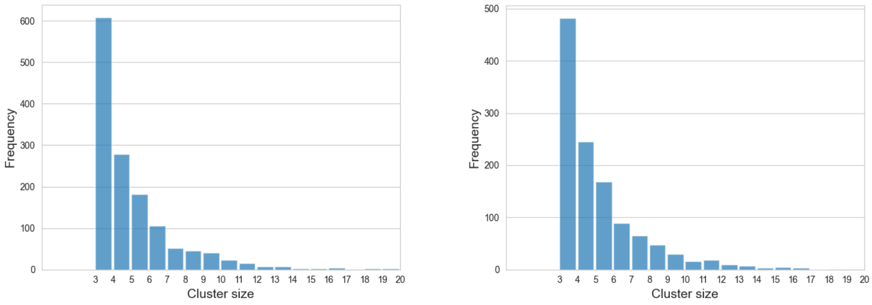
(a) Conditional coverage of M2M100$_{(1.2B)}$ for de → en.

(b) Conditional coverage of M2M100$_{(1.2B)}$ for ja → en.

(c) Conditional coverage for OPT$_{(350M)}$ on Language Modelling.

(d) Conditional coverage for OPT$_{(1.3B)}$ on Language Modelling.

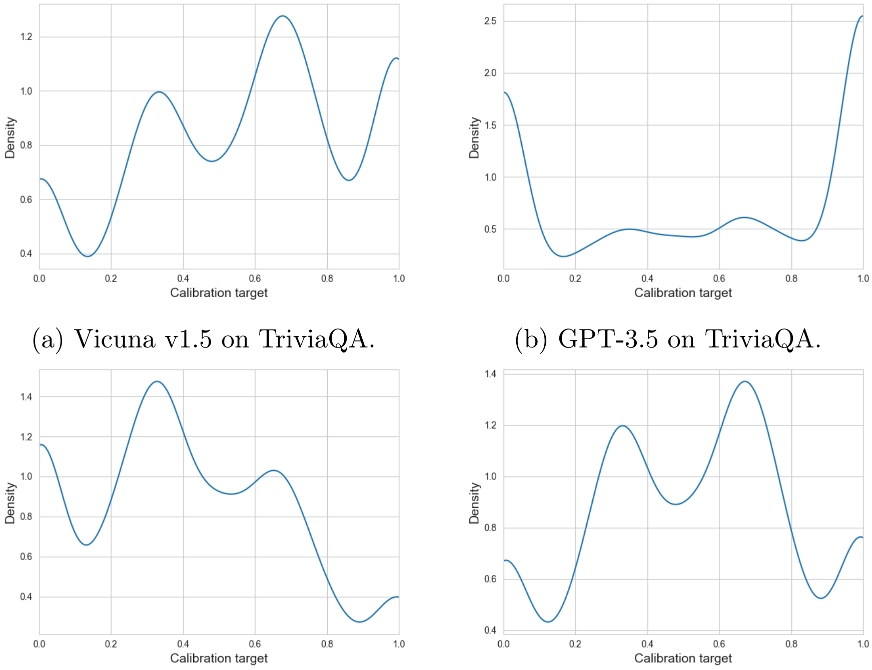
Figure B.17: Additional conditional coverage plots for the MT and LM dataset using our non-exchangeable conformal prediction method, aggregating predictions by prediction set size. The blue curve shows the conditional coverage per bin, whereas red bars show the number of predictions per bin. For Figures B.17c and B.17d, we zoom in on the prediction set sizes from 1 and 100.



(a) Cluster sizes on TriviaQA.

(b) Cluster sizes on CoQA.

Figure B.18: Bar plot of cluster sizes found. The plot is truncated at size 20.



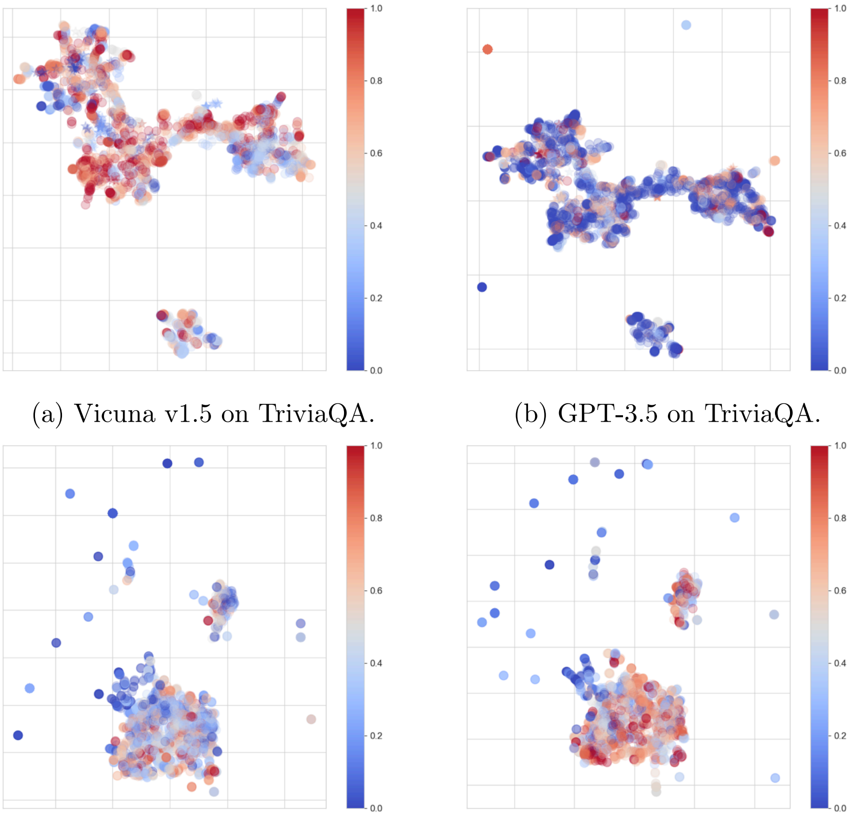
(a) Vicuna v1.5 on TriviaQA.

(b) GPT-3.5 on TriviaQA.

(c) Vicuna v1.5 on CoQA.

(d) GPT-3.5 on CoQA.

Figure B.19: Density plot of calibration targets generated through the clustering procedure for the two LLMs and TriviaQA / CoQA.



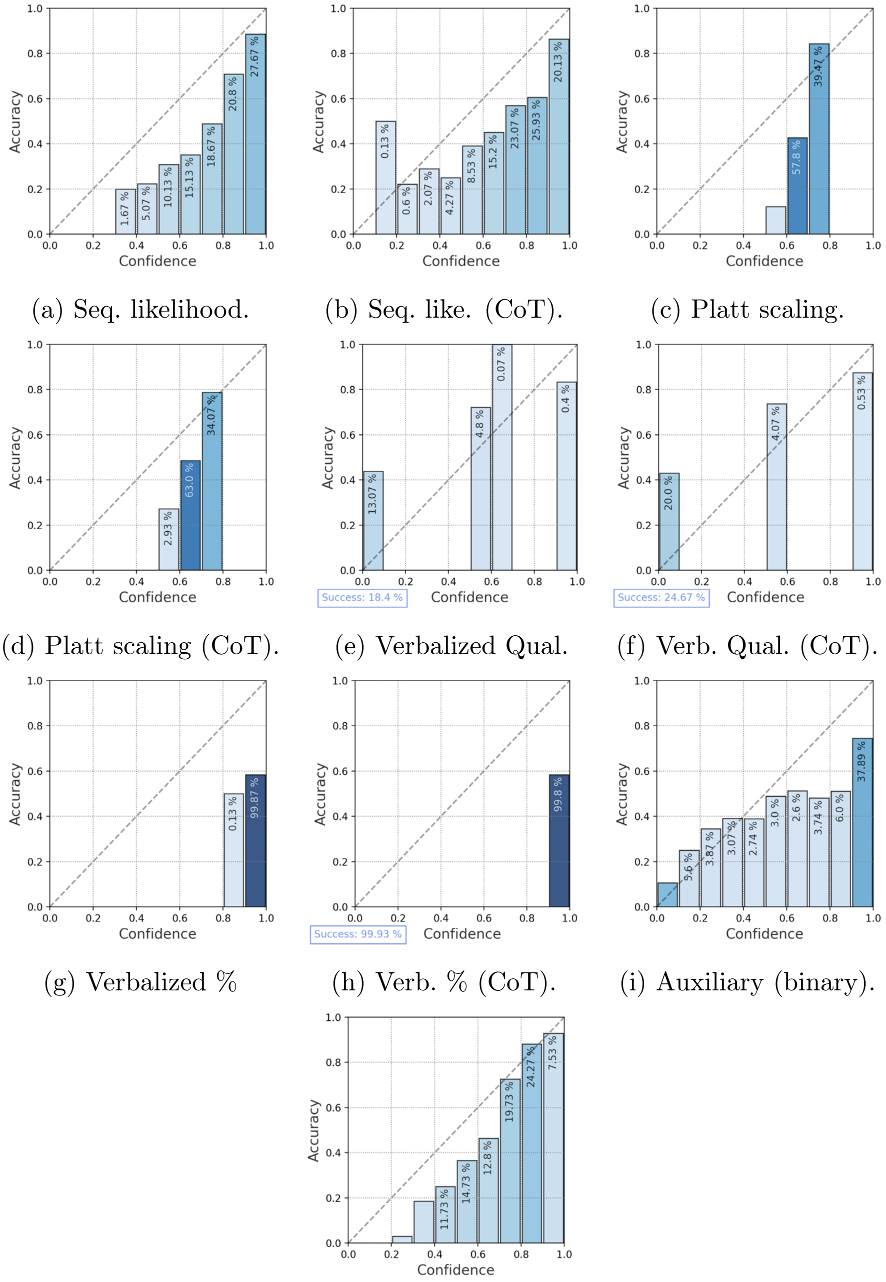
(a) Vicuna v1.5 on TriviaQA.

(b) GPT-3.5 on TriviaQA.

(c) Vicuna v1.5 on CoQA.

(d) GPT-3.5 on CoQA.

Figure B.20: Illustrating questions from TriviaQA along with their assigned confidence targets for the two LLMs, signified through their color from dark blue (0) to dark red (1). To avoid clutter, we subsampled 40% of the combined datasets to be shown here and used PacMAP (Wang et al., 2021b) to transform their sentence embeddings into 2D space.



(a) Seq. likelihood.

(b) Seq. like. (CoT).

(c) Platt scaling.

(d) Platt scaling (CoT).

(e) Verbalized Qual.

(f) Verb. Qual. (CoT).

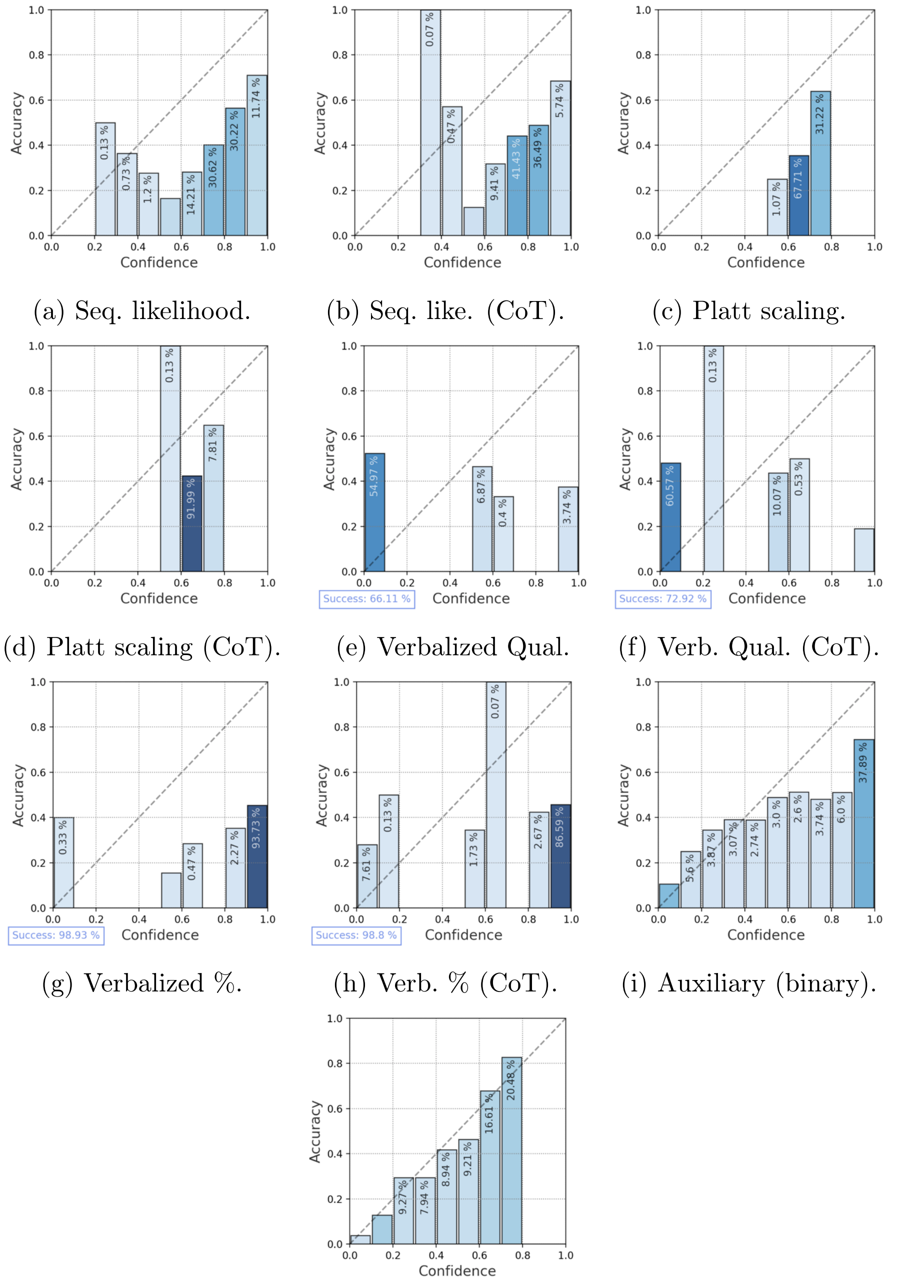
(g) Verbalized %

(h) Verb. % (CoT).

(i) Auxiliary (binary).

(j) Aux. (clustering).

Figure B.21: Reliability diagrams for all methods using 10 bins each for Vicuna v1.5 7B on TriviaQA. The color as well as the percentage number within each bar indicate the proportion of total points contained in each bin.



(a) Seq. likelihood.

(b) Seq. like. (CoT).

(c) Platt scaling.

(d) Platt scaling (CoT).

(e) Verbalized Qual.

(f) Verb. Qual. (CoT).

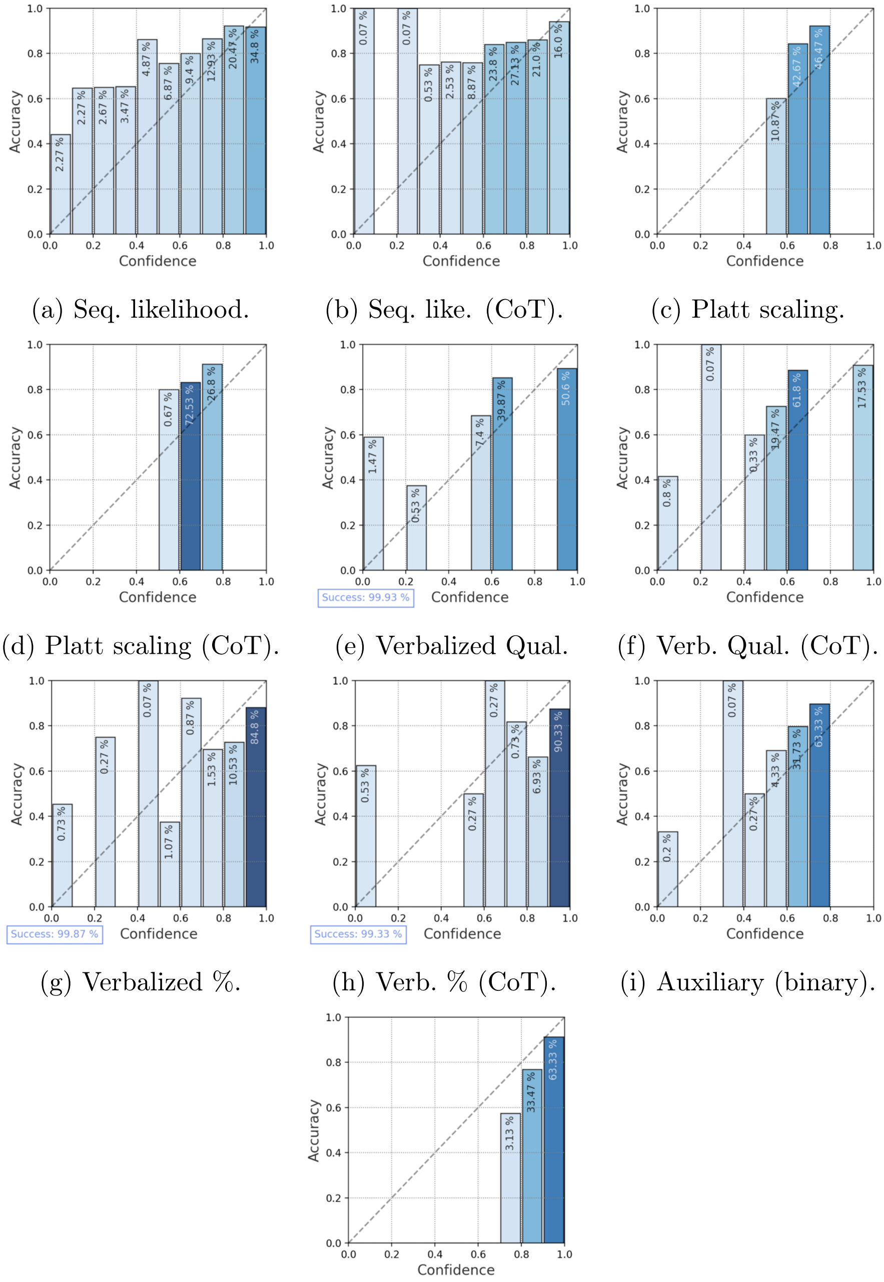
(g) Verbalized %.

(h) Verb. % (CoT).

(i) Auxiliary (binary).

(j) Aux. (clustering).

Figure B.22: Reliability diagrams for all methods using 10 bins each for Vicuna v1.5 7B on CoQA. The color as well as the percentage number within each bar indicate the proportion of total points contained in each bin.



(a) Seq. likelihood.

(b) Seq. like. (CoT).

(c) Platt scaling.

(d) Platt scaling (CoT).

(e) Verbalized Qual.

(f) Verb. Qual. (CoT).

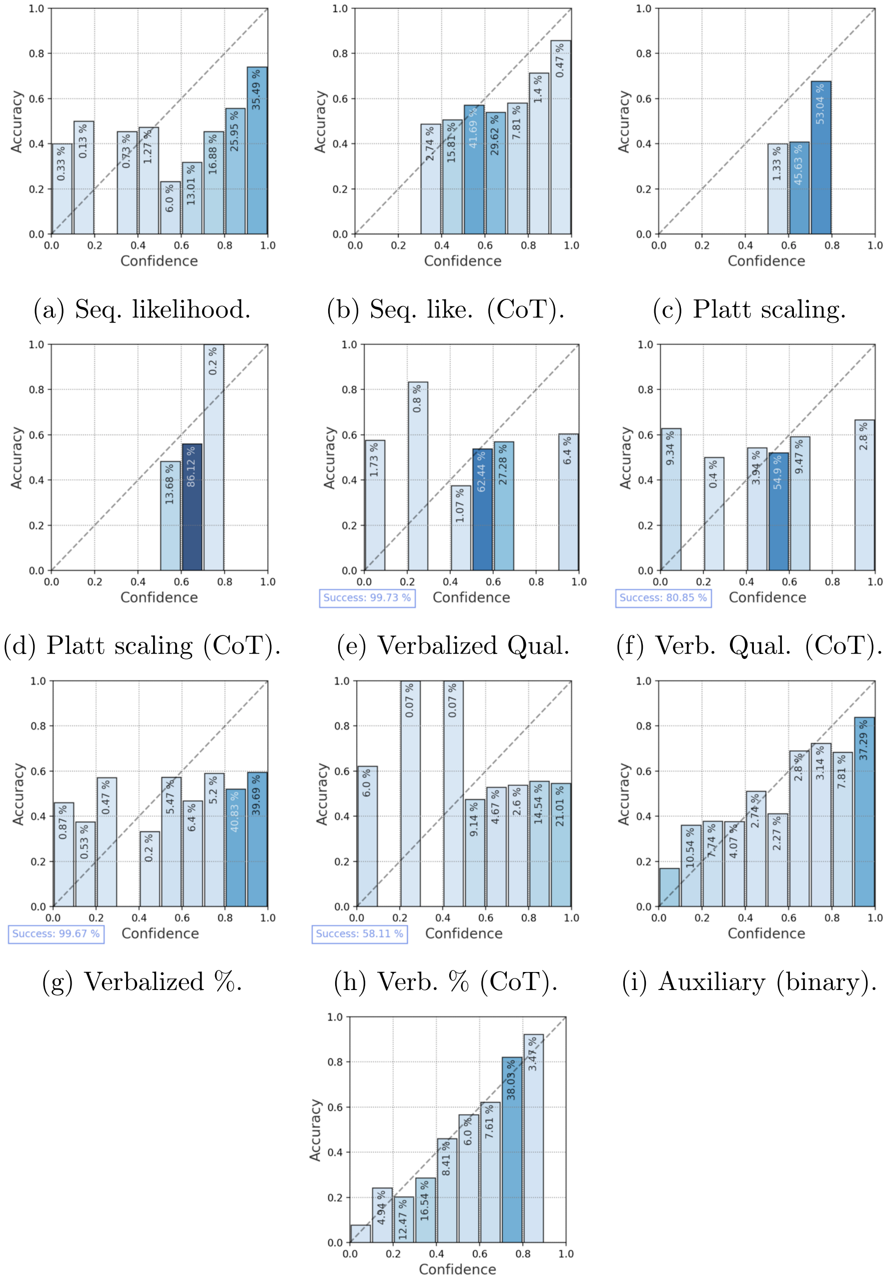
(g) Verbalized %.

(h) Verb. % (CoT).

(i) Auxiliary (binary).

(j) Aux. (clustering).

Figure B.23: Reliability diagrams for all methods using 10 bins each for GPT-3.5 on TriviaQA. The color as well as the percentage number within each bar indicate the proportion of total points contained in each bin.



(a) Seq. likelihood.

(b) Seq. like. (CoT).

(c) Platt scaling.

(d) Platt scaling (CoT).

(e) Verbalized Qual.

(f) Verb. Qual. (CoT).

(g) Verbalized %.

(h) Verb. % (CoT).

(i) Auxiliary (binary).

(j) Aux. (clustering).

Figure B.24: Reliability diagrams for all methods using 10 bins each for GPT-3.5 on CoQA. The color as well as the percentage number within each bar indicate the proportion of total points contained in each bin.

C | Reproducibility Appendix

> "*grad student descent: (machine learning, humorous) The process of choosing hyperparameters manually and in an ad-hoc manner, typical of work assigned to a graduate student.*"
>
> —Wiktionary definition.

Thesis	Appendix
Section 3.2.1	Appendix C.3
Section 2.2.3	Appendix C.4.1
Section 4.2.2	Appendix C.5
Section 4.2.1	Appendix C.6
Section 4.2.5	Appendix C.7
Section 5.4.1	Appendix C.8

Table C.1: Correspondences between sections of the reproducibility appendix and thesis chapters.

This appendix contains additional information for reproducibility purposes, according to the guidelines by Ulmer et al. (2022a). In Appendix C.1, a number of open-source software projects that were used in the making of this thesis are listed. Appendix C.2 discusses the compute hardware and environmental impact of the conducted experiments and other aspects. In Table C.1, we give an overview over the correspondence between thesis chapters and sections in this appendix.

C.1 Open Source Software

This thesis would not have been possible without the usage of open-source tools and software. Deep learning models where implemented with `NumPy` (Harris et al., 2020), `SciPy` (Virtanen et al., 2020), `scikit-learn` (Pedregosa et al., 2011), `einops` (Rogozhnikov, 2022), `PyTorch` (Paszke et al., 2019) and the `transformers` library (Wolf et al., 2020). Experimental tracking and hyperparam-

Repository	Chapters
github.com/Kaleidophon/phd-thesis	Sections 1.3, 2.1.2, 2.2.4 and 3.2.1
github.com/Kaleidophon/evidential-deep-learning-survey	Section 2.2.3
github.com/Kaleidophon/awesome-experimental-standards-deep-learning	Chapter 3
github.com/Kaleidophon/evidential-deep-learning-survey	Section 3.2.1
github.com/Kaleidophon/know-your-limits	Section 4.1
github.com/Kaleidophon/nlp-low-resource-uncertainty github.com/Kaleidophon/nlp-uncertainty-zoo	Section 4.2
github.com/Kaleidophon/non-exchangeable-conformal-language-generation	Chapter 5
github.com/parameterlab/apricot	Chapter 6

Table C.2: List of open-source repositories for the contents of this thesis.

eter search was facilitated via Weights & Biases (Biewald, 2020), and tracking carbon emissions with `codecarbon` (Schmidt et al., 2021; Lacoste et al., 2019; Lottick et al., 2019). The code for the plots and experiments in this thesis is itself available open-source, and corresponding online repositories are listed in Table C.2.

C.2 Environmental Impact

Here, we discuss the environmental impact of the experiments in the different chapters. In all cases, carbon emissions have been estimated using `codecarbon` (Schmidt et al., 2021; Lacoste et al., 2019; Lottick et al., 2019), although it should be noted that since the time of writing, more advanced tools like `LLMCarbon` (Faiz et al., 2023) have been developed to more accurately estimate the carbon footprint of language model training, specifically.

For Chapter 4, the carbon efficiency was estimated to be 0.61 kgCO$_2$eq / kWh. 735 hours of computation were performed on a Tesla V100 GPU. This includes hyperparameter search, failed runs, debugging, and discarded runs. As a rough upper bound, we estimate the compute time for a single replication of all experiments in Chapter 4 to take around 73 hours.[89] Total emissions were estimated to be 52.45 kgCO$_2$eq.

For Chapter 5, the carbon efficiency was estimated to be 0.12 kgCO$_2$eq / kWh. 159.5 hours of computation were performed on a NVIDIA RTX A6000. Total emissions are estimated to be 6.99 kgCo2eq. All of these values are upper bound including debugging as well as failed or redundant runs, and thus any replication of

[89] Note that this number could be reduced further by using better hardware acceleration, larger batch sizes, and slightly reducing the training duration for some models. Most importantly, this number also includes compute used for hyperparameter search.

results will likely be shorter and incur fewer carbon emissions.

For Chapter 6, all experiments were run on a single V100 NVIDIA GPU. We estimate finetuning the auxiliary calibrator to amount to 0.05 kgCO_2eq of emissions, with an estimated carbon efficiency of 0.46 kgCO_2eq / kWH. Therefore, we estimate total emissions of around 1 kgCO_2eq to replicate all the experiments in this chapter.

Carbon Offsetting. Carbon offsetting is a controversial topic (Watt, 2021; Campbell, 2021; Baras, 2023), and avoiding emission should always be the preferred option compared to post-hoc offsetting. Nevertheless, the author believes in mitigating the impact of their emissions as best as possible. The tracked carbon emissions from all the chapter in this thesis are 60.44 kgCO_2eq. An additional 20% is added to this number to account for variation in tracking, untracked debug runs or failed experiments, amounting to 72.53 kgCO_2eq. Furthermore, over the course of almost four years, the author attended a number of conferences during their PhD program, and conducted industrial internships as well as a research stay. The travels related to these activities produced an estimated total of 12088 kgCO_2eq in emissions. Direct air capture by climeworks (climeworks, 2022) was used to offset the emissions from the experiments, and carbon credits stemming from wind energy projects in Thailand and India were purchased through the Gold Standard Marketplace (Gold Standard, 2024) for travel-related emissions.

C.3 ASO Test Implementation Details

This section details the Python implementation of the ASO test in Section 3.2.1. The full algorithm to compute the $\varepsilon_{\min}$ score is given in Algorithm 3, and will now be explained in full detail. We show how the violation ratio in Equation (3.3) can be compute in Python:

```
def compute_violation_ratio(scores_a: np.array, scores_b:
↪  np.array, dt: float) -> float:
    quantile_func_a = get_quantile_function(scores_a)
    quantile_func_b = get_quantile_function(scores_b)

    t = np.arange(dt, 1, dt)  # Points we integrate over
    f = quantile_func_a(t)  # F-1(t)
    g = quantile_func_b(t)  # G-1(t)
    diff = g - f
```

Algorithm 3 Almost Stochastic Order (ASO) Significance Test

```
Require: Sets of observations $\mathbb{S}_\mathbb{A}$ and $\mathbb{S}_\mathbb{B}$, integration interval $\Delta_t$,
  number of bootstrap iterations $B$, desired confidence level $1-\alpha$.
  $\varepsilon_{\mathcal{W}_2}(F_n, G_m)$ = compute_violation_ratio($\mathbb{S}_\mathbb{A}$, $\mathbb{S}_\mathbb{A}$, $\Delta_t$)

  ▷ Bootstrapping
  for $i \in\ 0, \ldots, B$ do
      $\mathbb{S}^*_\mathbb{A}$ = bootstrap_sample($\mathbb{S}_\mathbb{A}$)

      $\mathbb{S}^*_\mathbb{B}$ = bootstrap_sample($\mathbb{S}_\mathbb{B}$)

      ▷ Store value below in list
      $\varepsilon^*_{\mathcal{W}_2}(F_n, G_m)$ = compute_violation_ratio($\mathbb{S}^*_\mathbb{A}$, $\mathbb{S}^*_\mathbb{A}$, $\Delta_t$)

  end for
  ▷ Compute value below based on variance of all the values in list
  $\hat{\sigma}^2_{n,m} = \mathrm{Var}\left[\sqrt{\frac{mn}{n+m}}\left(\varepsilon_{W_2}(F^*_n, G^*_m) - \varepsilon_{W_2}(F_n, G_m)\right)\right]$
  $\varepsilon_{\min}(F_n, G_m, \alpha) = \varepsilon_{W_2}(F_n, G_m) - \sqrt{\frac{n+m}{nm}}\hat{\sigma}_{n,m}\Phi^{-1}(\alpha)$
  return $\varepsilon_{\min}(F_n, G_m, \alpha)$
```

```
    squared_wasserstein_dist = np.sum(diff ** 2 * dt)

    # Now only consider points where stochastic order is being
    ↪  violated and set the rest to 0
    diff[f >= g] = 0
    int_violation_set = np.sum(diff[1:] ** 2 * dt)  # Ignore t =
    ↪  0 since t in (0, 1)

    violation_ratio = int_violation_set /
    ↪  squared_wasserstein_dist

    return violation_ratio
```

We can see that the integration over the violation set $\mathbb{V}_X$ in Equation (3.3) is being performed by masking out values for which the stochastic order is honored (i.e. where $F_n^{-1}(t) \geq G_n^{-1}(t)$). Computing the violation ratio involves building the empirical inverse cumulative distribution function or empirical quantile function, the same method as in Dror et al. (2019) is used, with the corresponding Python code given below:

```
def get_quantile_function(scores: np.array) -> Callable:
    def _quantile_function(p: float) -> float:
        cdf = np.sort(scores)
        num = len(scores)
        index = int(np.ceil(num * p))

        return cdf[np.clip(index - 1, 0, num - 1)]

    return np.vectorize(_quantile_function)
```

This function is also used inside the bootstrap sampling procedure, the last missing part of the implementation. We again follow the implementation by Dror et al. (2019) and employ the inverse transform sampling procedure, in which we draw $p \sim \mathcal{U}[0,1]$ and run it through a quantile function to create a sample.

C.4 Hyperparameters Search

Here we list the hyperparameter search procedures, ranges and found values for the different experiments in this thesis, in the order of appearance.

C.4.1 Iris Example

This section describes the details for the Iris dataset example in Figure 2.8 in Section 2.2.3. All models use three layers with 100 hidden units and ReLU activations each. We furthermore optimized all of the models with a learning rate of 0.001 using the Adam optimizer (Kingma and Ba, 2015) with its default parameter settings. We also regularize the ensemble and MC Dropout model with a dropout probability of 0.1 each.

Prior Network Specifics. We choose the expected l_2 loss by Sensoy et al. (2018) and regularize the network using the KL divergence w.r.t. to a uniform Dirichlet as in Sensoy et al. (2018). In the regularization term, we do not use the original concentration parameters $\boldsymbol{\alpha}$, but a version in which the concentration of the parameter α_k corresponding to the correct class is removed using a one-hot label encoding $\mathbf{y}$ by $\tilde{\boldsymbol{\alpha}} = (1 - \boldsymbol{\alpha}) \circ \boldsymbol{\alpha} + \mathbf{y} \circ \boldsymbol{\alpha}$. The regularization term is added to the loss using a weighting factor of 0.05.

C.4.2 Synthetic Data Experiments

This sections gives more details on the synthetic data experiments in Section 4.1.5. We perform our experiments on the half-moons dataset, using the corresponding function to generate the dataset in `scikit-learn` (Pedregosa et al., 2011), producing 500 samples for training and 250 samples for validation using a noise level of .125. We do hyperparameter search using the ranges listed in Table C.4, settling on the values given in Table C.3 after 200 evaluation runs per model (for neural networks and MC dropout; the hyperparameters found for neural networks were then used for Platt scaling, anchored ensembles and neural ensembles as well). We also performed a similar hyperparameter search for the Bayes-by-backprop (Blundell et al., 2015) model, which seemed to not have yielded a suitable configuration even after extensive search, which is why results were omitted here. All models were trained with a batch size of 64 and for 20 epochs at most using early stopping with a patience of 5 epochs and the Adam optimizer.

Model	Hyperparameter	Value
Neural Network	Hidden size	$[25, 25, 25]$
Neural Network	Dropout prob.	.014552
Neural Network	Learning rate	.000538
MC Dropout	Hidden sizes	$[25, 25, 25, 25]$
MC Dropout	Dropout prob.	.205046
MC Dropout	Learning rate	.000526

Table C.3: Best hyperparameters found on the half-moon dataset.

Hyperparameter	Chosen from
Hidden layers	1–5 layers of 15, 20, 25
Learning rate	$\mathcal{U}(\log 10^{-4}, \log 0.1)$
Dropout rate	$\mathcal{U}(0.1, 0.5)$

Table C.4: Distributions or options that hyperparameters were sampled from during the random hyperparameter search.

C.4.3 Text Classification Experiments

Here, we detail the hyperparameter search conditions for the experiments in Section 4.2.5. We perform hyperparameter search using random sampling (Bergstra and Bengio, 2012) using hyperband scheduling (Li et al., 2017)[90] on the entire training set, even if models are trained on sub-sampled training sets later. This has the advantage of ensuring comparability between runs and eliminating suboptimal hyperparameter choices as a source of worse uncertainty estimation. We do 80 trials for LSTM-based models,

[90] Trials might be terminated using hyperband after $10k$ steps.

and 30 for Bert-based models. Furthermore, the hyperparameters for the LSTM are identical for the LSTM ensemble (10 instances are used per ensemble). Hyperparameters were picked by best final validation loss over search trials.

Chosen Hyperparameters. We summarize some common hyperparameters here and show the rest in Table C.6. We commonly use a batch size of 32, and sequence lengths of 35 for LSTM-based and 128 for Bert-based models. All LSTM-based models are trained using 2 layers, with the exception of the vanilla LSTM and the LSTM-ensemble on Clinc Plus with 3 layers. Their hidden size and embedding sizes are set to 650. For all models, gradient clipping is set to 10. For models using multiple predictions to compute uncertainty estimates, 10 predictions are used at a time.

Name	Tuned for	Search space
Learning rate	LSTM, LSTM Ensemble, Bayesian LSTM, ST-τ LSTM Variational LSTM	$\mathcal{U}(0.1, 0.5)$
Learning rate	DDU BERT, SNGP BERT, Variational BERT	$\log\mathcal{U}(10^{-5}, 10^{-3})$
Spectral norm upper bound	DDU BERT, SNGP BERT	$\mathcal{U}(0.95, 0.99)$
Kernel amplitude	SNGP BERT	$\log\mathcal{U}(0.01, 0.5)$
β weight decay	SNGP BERT	$\log\mathcal{U}(10^{-3}, 0.5)$
Weight decay	LSTM, LSTM Ensemble, ST-τ LSTM, Variational BERT	$\mathcal{U}(0.1, 0.5)$
Layers	LSTM, LSTM Ensemble	$\{2, 3\}$
Dropout	LSTM, LSTM Ensemble, ST-τ LSTM, Variational BERT	$\mathcal{U}(0.1, 0.4)$
Layer Dropout	Variational LSTM	$\mathcal{U}(0.1, 0.4)$
Time Dropout	Variational LSTM	$\mathcal{U}(0.1, 0.4)$
Embedding Dropout	Variational LSTM	$\mathcal{U}(0.1, 0.4)$
Hidden size	LSTM, LSTM Ensemble	$\{350, 500, 650\}$
Prior σ_1	Bayesian LSTM	$\log\mathcal{U}(-0.8, 0.1)$
Prior σ_2	Bayesian LSTM	$\log\mathcal{U}(-0.8, 0.1)$
Prior π	Bayesian LSTM	$\log\mathcal{U}(0.1, 0.9)$
Posterior μ init	Bayesian LSTM	$\mathcal{U}(-0.6, 0.6)$
Posterior ρ init	Bayesian LSTM	$\mathcal{U}(-8, -2)$
Init weight	LSTM	$\mathcal{U}(0.1, 0.4)$
Number of centroids	ST-τ LSTM	$\{5, 10, 20, 30, 40\}$

Table C.5: List of searched hyperparameters. LSTM Ensemble hyperparameters are not searched, but simply copied from the found LSTM hyperparameters.

We further include some notes about the optimization of models for the experiments in Section 4.2.5. To make sure that all models are trained for the same number of steps regardless of the the size of (sub-sampled) training set, we set the training duration to the number of steps corresponding to a number of epochs using the original training set size, and name it *epoch-equivalents* in the following. Due to the imbalance of classes in Finnish UD and Dan+, all models were trained using loss-weights that are inverse to the frequency of a label in the dataset.

Optimization of LSTMs. We adopt different optimization schemes for transformer- and LSTM-based models. For LSTMs, we choose stochastic gradient descent with a decaying learning rate schedule, decaying by .8695 after the equivalent of 14 epochs for every following epoch-equivalent for 55 epoch-equivalents in total. This corresponds to the setup in Gal and Ghahramani (2016a), modified from the setup in Zaremba et al. (2014).

Optimization of Berts. We fine-tune Bert models using the shorter duration of 20 epoch-equivalents, corresponding to the NLP experiments in Liu et al. (2023). Adam (Kingma and Ba, 2015) is used for optimization with default parameters $\beta_1 = .9$ and $\beta_2 = .999$ alongside a triangular learning rate, using the first 10% of the training duration as warm-up.

C.4.4 Auxiliary Calibrator Experiments

This sections explains the hyperparameter tuning for the experiments in Section 6.2.2. We conduct suites of hyperparameter searches per target LLM, dataset and type of calibration targets (binary and clustering) corresponding to the results in Table 6.3, resulting in eight different suites. We then use these found hyperparameters for the results in Table 6.5.

Search Method and Ranges. For the search, we opt for Bayesian hyperparameter search (Snoek et al., 2012) as implemented by Weights & Biases (Biewald, 2020). We optimize only two hyperparameters: Learning rate and weight decay. The learning rate is samples from a log-uniform distribution log $\mathcal{U}[10^{-5}, 0.01]$ and weight decay from a uniform distribution $\mathcal{U}[10^{-4}, 0.05]$ for a total of 50 runs and 250 training steps each. The final hyperparameters selected are given in Table C.7.

Other Hyperparameters. When obtaining the responses from Vicuna v1.5 7B, we use a batch size of 4 and generate for a maximum of 50 tokens and stop generation when the model tries to generate parts of the prompt, such as “Question:” / “Q:” or “Answer:” / “A:”. We also use 10 in-context samples for TriviaQA, but no in-context samples for CoQA. For the auxiliary calibrator, we use a context size of 512 tokens, batch size of 32, and gradient clipping with a maximum norm of 10.

C.5 Pre-processing for Text Classification Benchmark

This sections explains the data preprocessing for the datasets used in Section 4.2.5.

Tokenization. We use the corresponding Bert tokenizer for each language, including for LSTM-based models to ensure compatibility. For English, this corresponds to the original SentencePiece tokenizer used by Devlin et al. (2019), while we use the tokenizer of the Danish Bert (Hvingelby et al., 2020) and Finnish Bert (Virtanen et al., 2019) for those languages, respectively.

Tags for Sub-Word Tokens. For named entity recognition and part-of-speech tagging, we follow Jurafsky and Martin (2022), chapter 11.3.3 to deal with sub-word tokens: For every token that is split into sub-word tokens, we assign the tag only to the first sub-word token, and -100 for the rest, which ignores them for evaluation purposes.

C.6 Implementation Details of Text Classification Benchmark

This section gives additional implementation details for the models used in the text classification benchmark in Section 4.2.5.

Resources. In addition to the resources in Appendix C.1, the Bayesian LSTM was developed using the `Blitz` package (Esposito, 2020) for PyTorch and the SNGP transformer using `gpytorch` (Gardner et al., 2018).

Models. For the DUE transformer, we used principal component analysis on the latent representations for Clinc Plus to reduce the memory usage of the Gaussian discriminant analysis by reducing dimensionality to 64. We initially also experimented with the usage of the DUE transformer by van Amersfoort et al. (2021), however found that it was not trivial to create the inducing points for the

Gaussian process output layer in a sequential setting. For the variational transformer (Xiao et al., 2020), the authors do not specify exactly how MC dropout is used. We use the existing dropout layers in the corresponding model, and use a number of forward passes with different dropout masks to make predictions. Since the number of classes is prohibitive for the original formulation of the SNGP transformer, we use the extension proposed by Liu et al. (2023) in Appendix A.1 and only store one $\hat{\Sigma}^{-1}$ matrix for all classes. Furthermore, we update the matrix continuously during training and not just during the last epoch, in order to allow tracking of the predictive performance over the training time. Lastly, we also evaluate predictions using Monte Carlo approximations instead of the mean-field approach, since this allows us to compute a wider variety of uncertainty metrics.

Evaluation. When computing uncertainty estimates and losses for evaluation purposes, the measurements for a number of tokens were discarded. These include the ignore token with ID -100, as well as the IDs corresponding to the `[EOS]`, `[SEP]`, `[CLS]` and `[PAD]` token, which might differ between tokenizers of different languages. For computing the ECE, we use 10 bins.

Model Comparison. We facilitate the comparison of models using the almost stochastic order test from Section 3.2.1. We use the test with a confidence level $\alpha = 0.05$ and a decision threshold of $\tau = 0.3$.

C.7 Convergence on Clinc Plus

Here, we briefly address the models missing from the English Clinc Plus experiments in Section 4.2.5. For the ST-τ and variational LSTM, we could not identify clear reasons on why models did not converge. Even after extensive hyperparameter searches and manual fine-tuning of hyperparameters (including different learning rate schedules and optimizers), we did not find a combination of options that resulted in convergence. We also observed strange behavior for the Bayesian LSTM, which, after reaching a validation accuracy of 0.5, would suddenly return to its initial training performance. This could potentially be explained by the model accidentally escaping a low-loss basin due to a learning rate that is still too high, and thus we changed the model to only be trained for 18 epoch-equivalents and initiate the learning rate decay after seven epoch-equivalents. The puzzling fact is that SNGP Bert did not converge on Clinc Plus, since the authors successfully used the dataset in their own work (Liu et al., 2023). We put forth the following explanations: First of all, we observed the model to generally possess a high variance, as demonstrated by the standard deviation on the Danish and Finnish data. Secondly, we make at least two changes to their implementation: Instead of using the mean-field approximation to the predictive distribution, we use the Monte Carlo approximation in order to compute metrics such as mutual information. Also, we update the covariance matrix $\hat{\Sigma}$ over the whole training time in order to track the predictive performance for our experiments, and not just during the last epoch.

C.8 Temperature Search

This sections explains the temperature search procedure for the parameter τ in Equation (5.5) in Section 5.3 further. To determine the temperaturein Equation (5.5) for the different distance metrics in Table 5.1, we adopt a variation of a simple hill-climbing procedure. Given user-defined bounds for the temperature search τ_{min} and τ_{max}, we sample an initial candidate $\tau_0 \sim \mathcal{U}[\tau_{\text{min}}, \tau_{\text{max}}]$, and then evaluate the coverage of the method given the candidate on the first 100 batches of the calibration dataset. The next candidate then is obtained via

$$\begin{aligned} \tau_{t+1} &= \tau_t + \eta \cdot \varepsilon \cdot \text{sgn}\big(1 - \alpha - \text{Coverage}(\tau_t)\big); \\ \varepsilon &\sim \mathcal{N}(0, \tau_{\text{max}} - \tau_{\text{min}}), \end{aligned} \tag{C.1}$$

where η is a predefined step size (in our case 0.1) and Coverage(τ_t) the achieved coverage given a candidate τ_t. The final temperature is picked after a fixed number of steps ($t = 20$ in our work) based on the smallest difference between achieved and desired coverage.

Overall, we found useful search ranges to differ greatly between experimental settings, as illustrated by the reported values in Table 5.1 and Table 5.2. In general, the stochastic hill-climbing could also be replaced by a grid search, even though we sometimes found the best temperature to be "hidden" in a very specific value range. It also has to be noted that temperature for the l_2 distance is the highest by far since FAISS returns *squared* l_2 distances by default.

Model	Hyperparameter	English	Danish	Finnish
LSTM	Weight decay	.001337	.001357	.001204
	Learning rate	.4712	.4931	.2205
	Init. weight	.2830	.5848	.5848
	Dropout	.3379	.2230	.1392
Variational LSTM	Weight decay	–	10^{-7}	.01953
	Learning rate	–	.3031	.7817
	Init. weight	–	.1097	.5848
	Embedding Dropout	–	.1207	.3566
	Layer Dropout	–	.1594	.3923
	Time Dropout	–	.1281	.1646
Bayesian LSTM	Weight decay	.001341	.003016	.03229
	Learning rate	.1704	01114	.1549
	Dropout	.3410	.3868	.331
	Prior σ_1	.9851	.7664	.3246
	Prior σ_2	.5302	.851	.5601
	Prior π	1	1	.1189
	Posterior μ init	$-.005537$	$-.0425$	.4834
	Posterior ρ init	-7	-6	.1124
ST-τ LSTM	Weight decay	–	.001189	.0007857
	Learning rate	–	.01979	.3601
	Dropout	–	.1867	.1737
	Num. centroids	–	5	30
DDU Bert	Learning Rate	.003077	.00006168	.001825
	Spectral norm upper bound	.9753	.9211	.9410
	Weight decay	$.0039 = 0$	.1868	.09439
Variational BERT	Learning Rate	.0002981	.00009742	.00003483
	Weight decay	.01591	.02731	.09927
	Dropout	.2382	.4362	.4364
SNGP Bert	Learning Rate	–	.0002332	.0002919
	Spectral norm upper bound	–	.99	.96
	Beta Weight decay	–	.001619	.002438
	Beta length scale	–	2.467	2.254
	Kernel amplitude	–	.3708	.2466

Table C.6: List of used model hyperparameters by dataset.

		TriviaQA		CoQA	
		Binary	Clustering	Binary	Clustering
Vicuna v1.5	learning rate	1.4×10^{-5}	3.37×10^{-5}	9.58×10^{-5}	8.84×10^{-5}
	weight decay	.03184	.008936	.005793	7.42×10^{-4}
GPT-3.5	learning rate	2.96×10^{-5}	1.62×10^{-5}	5.12×10^{-5}	5.59×10^{-5}
	weight decay	.01932	.01362	.03327	.03495

Table C.7: Chosen hyperparameters for our model on different datasets and for different calibration targets.

Abbreviations

***k*-NN** k-nearest neighbors, the idea to use k points most similar to a point of interest for purposes such as classification or clustering.

AI Artificial intelligence, the field that develops and studies methods that enables machines to learn and take actions in a way that imitates human intelligence.

API Application programming interface, a way for computer programs to communicate with each other through a specified interfance.

APRICOT Auxiliary prediction of confidence targets. Method to create calibrated confidence scores for black-box LLMs through an external secondary model, discussed in Chapter 6.

AUPR Area under the precision-recall curve, an evaluation metric that measures the trade-off between the precision and recall under varying decision thresholds for a binary classification problem.

AUROC Area under the receiver-operator characteristic, an evaluation metric that measures the trade-off between the true positive rate and false positive rate under varying decision thresholds for a binary classification problem.

Bert Bidirectional encoder representations from transformers by Devlin et al. (2019), a transformer-based language trained to predict a masked-out token given some context, with predicting the order of two subsequent sentences as an auxiliary task.

BLEU Bilingual evaluation understudy, an evaluation metric initially proposed by Papineni et al. (2002) to evaluate machine translation methods based on the n-gram overlap between the translation and a reference.

COMET Crosslingual optimized metric for evaluation of translation, a metric proposed by Rei et al. (2020) that tries to predict the quality of a machine-generated translation using a neural model.

CoT Chain-of-through prompting, a prompting technique originally proposed by Wei et al. (2022), where an LLM is instructed to solve a task by performing step-by-step reasoning.

CP Conformal prediction, a technique orginally developed by Vovk et al. (2005) that creates prediction sets (classification) or intervals (regression) that include the correct prediction with a pre-defined probability, given that a test point is from the same distribution as the calibration data used to construct the prediction sets / intervals.

DDU Deep deterministic uncertainty transformer Mukhoti et al., 2021, a type of transformer for which a Gaussian discriminant analysis model is fit on its latent features in order to quantify uncertainty.

DL Deep learning, the field concerned with the study of artificial neural network of increased depth. Can be considered a subfield of machine learning.

FAISS (Meta's) Fundamental AI similarity search, a software library proposed by Johnson et al. (2019) to quickly find the nearest neighbors for a vector in a datastore.

HDBSCAN Hierarchical density-based spatial clustering of applications with noise (Campello et al., 2013), an improvement on the earlier DBSCAN clustering algorithm (Ester et al., 1996). The algorithm is unsupervised, i.e. does not require a specification of the number of clusters a priori, and works by merging points into cluster by distance in a bottom-up fashion.

LLM Large language model or foundation model; typically a large neural model based on the transformer architecture (Vaswani et al., 2017), that has been trained on huge swaths of data to model the statistical distribution of words.

LM Language model or language modeling (i.e. the process or task of modeling the statistical distribution of words underlying language). More general than LLMs, since language models can also be based on recurrent or n-gram models.

LSTM Long-short term memory network (Hochreiter and Schmidhuber, 1997), a type of recurrent neural architecture used for sequential data.

MAUVE Neural metric to assess the quality of machine-generated, general text by Pillutla et al. (2021).

ML Machine learning, the subfield of artificial intelligence concerned with the development and study of statistical algorithms that can learn from data and generalize to unseen data.

MT Machine translation, the study or task of automatically translating text or speech with the help of computers.

NLG Natural language generation, a set of tasks involving the generation of language, including language modeling, machine translation, question-answering, and image captioning.

NLP Natural language processing, an interdisciplinary subfield of artificial intelligence and linguistics, primarily concerned with providing computers the ability to process data encoded in natural language.

OOD Out-of-distribution or out-of-domain, used to refer to test inputs to a machine learning model that are different come from a different distribution than the training data the model was originally fit on.

PAC Probably approximately correct learning, a framework for the mathematical analysis of machine learning.

ReLU Rectifier linear unit, a non-linear activation function defined as $\phi(x) = \max(0, x)$, often used on the activations between neural network layers.

RLHF Reinforcement learning from human feedback (Christiano et al., 2017; Stiennon et al., 2020), a technique to optimize neural models based on human preference data.

ROUGE An evaluation metric initially proposed by (Lin, 2004) to text summarization methods based on the n-gram overlap between the summarization and a reference.

SDE Stochastic differential equation, a type of equation regarding the derivative of some function in which one or more of the terms is a stochastic process.

SNGP Spectrally-normalized Gaussian process transformer (Liu et al., 2023), a type of transformer architecture whose weights are regularized through spectral normalization and which features a Gaussian process output layer..

UQ Uncertainty Quantification; methods to assess the reliability or trustworthiness of the predictions of (in this thesis) neural models.

Index