Deep Generative Models for Faces and Expressions

René Haas

Machine Learning Research Group
Computer Science Department
IT University of Copenhagen

Principal Supervisor
Associate Professor Sami S. Brandt

Co-supervisor
Assistant Professor Stella Graßhof

This dissertation was submitted, September 30th, 2023, to the Computer Science Department in partial fulfillment of the requirements for the degree of Doctor of Philosophy (PhD) at the IT University of Copenhagen.

Defended in Copenhagen, Denmark, November 24th, 2023.

Examination committee

Associate Professor Veronika Cheplygina
IT University of Copenhagen

Samuli Laine
Distinguished Research Scientist, NVIDIA

Professor Fredrik Kahl
Chalmers University of Technology

Declaration of Work

I declare that this thesis - submitted in partial fulfilment of the requirements for the conferral of 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 in Denmark or abroad.

- *René Haas*

Abbreviations

3DMM 3D Morphable Model

AdaIN Adaptive Instance Normalization

BU-3DFE Binghamton University 3D Facial Expression

CFG Classifier-Free Guidance

CLIP Contrastive Language-Image Pretraining

CNN Convolutional Neural Network

DDIM Denoising Diffusion Implicit Model

DDM Denoising Diffusion Model

DDPM Denoising Diffusion Probabilistic Models

DGM Deep Generative Model

FFHQ Flickr-Faces-HQ

FID Fréchet Inception Distance

GAN Generative Adversarial Network

HOSVD Higher-Order Singular Value Decomposition

JVP Jacobian Vector Product

LDM Latent Diffusion Model

LPIPS Learned Perceptual Image Patch Similarity

MLP Multilayer Perceptron

NeRF Neural Radiance Field

NFE Neural Function Evaluation

NRSfM Non-Rigid Structure-from-Motion

PCA Principal Component Analysis

PPL Perceptual Path Length

ProGAN Progressive GAN

SfM Structure-from-Motion

SVD Singular Value Decomposition

SVM Support Vector Machine

VAE Variational Autoencoder

WGAN Wasserstein GAN

Acknowledgements

This PhD thesis represent the culmination of the work conducted during my three-year employment as a PhD fellow at the IT university of Copenhagen. The work focuses on developing methods for achieving greater control over face images produced by deep generative models. There are many people I would like to thank for accompanying me though my PhD journey. It is because of all of you that this dissertation exists at all.

First, I would like to express my sincere gratitude to my academic advisers Sami Sebastian Brandt and Stella Graßhof for their continuous support, guidance, encouragements and patience though the past three years. Without you I would not be where I am today. Further, I would like to thank Riko Jacob, Dan Witzner Hansen and Leon Derczynski for agreeing to be on my midway evaluation committee and for the interesting discussion we had and the constructive feedback I received during the meeting.

I am especially grateful to Tomer Michaeli for agreeing to host and advice me during my research stay abroad at the Technion - Israel Institute of Technology. I want to express my sincere appreciation to all members in Tomer Michaeli's group for making me feel so welcomed and for our many academic discussions. Finally, my appreciation and gratitude also go out to my cohort of fellow PhD students at ITU. Especially I would like to thank my PhD Colleagues Laura Weihl, Adrian Hoff, Magnus Ibh, Meisam Seikavandi and Mike Graham for your friendship and support during my journey.

English Abstract

This PhD thesis proposes several novel methods for semantic editing of human faces using Deep Generative Models (DGMs). DGMs such as Generative Adversarial Networks (GANs) and Denoising Diffusion Models (DDMs) have seen rapid improvements in image quality in recent years and are now able to synthesize human face portraits that are almost indistinguishable from real photographs. DGMs compress the high-dimensional manifold of plausible face images to a reduced representation called the latent space. This thesis seeks to develop methods for discovering preferred directions or trajectories in the latent space of DGMs that correspond to semantically interpretable changes to the generated images, for example, changes to pose or facial expression. The first part of the thesis focuses on StyleGAN, a state-of-the-art GAN architecture that has revolutionized the field of unconditional synthesis of human faces. First, this thesis proposes a novel editing method for StyleGAN using a multilinear tensor model and a real facial expression data set as supervision. Next, this thesis explores how StyleGAN represents 3D structure. StyleGAN generates 2D images and does not have any explicit 3D understanding. We use Non-Rigid Structure-from-Motion (NRSfM) to recover the underlying 3D structure from generated 2D images and propose a novel method for connecting the NRSfM model with the latent space of StyleGAN, allowing for explicit control of the 3D geometry of the generated images. Very recently, DDMs have emerged as a strong competitor to GANs, both in terms of the quality and diversity of the generated images. However, the latent space of DDMs is still not well understood. The final part of this thesis proposes novel supervised and fully unsupervised approaches for semantic editing of face images using DDMs.

Resume på Dansk

Denne PhD afhandling foreslår flere nye metoder til semantisk redigering af ansigter ved hjælp af Deep Generative Models (DGMs). DGMs såsom Generative Adversarial Networks (GANs) og Denoising Diffusion Models (DDMs) har set hurtige forbedringer i billedkvalitet de seneste år og er nu i stand til at syntetisere ansigter, der næsten er umulige at skelne fra ægte fotografier. DGMs komprimerer den højdimenionale manifold af plausible ansigts biller til en reduceret repræsentation kaldet det latente rum. Denne afhandling søger at udvikle metoder til at opdage foretrukne retninger eller baner i det latente rum DGMs', der svarer til semantisk fortolkelige ændringer i de genererede billeder, for eksempel ændringer i positur eller ansigtsudtryk. Første del af afhandlingen fokuserer på StyleGAN, en state-of-the-art GAN-arkitektur, der har revolutioneret feltet indenfor syntese af ansigtsbilleder. Først foreslår denne afhandling en ny redigeringsmetode for StyleGAN ved hjælp af en multilinear tensor model og et ansigtsudtryks datasæt som supervision. Derefter udforsker denne afhandling, hvordan StyleGAN repræsenterer 3D-struktur. StyleGAN genererer 2D-billeder og har ikke nogen eksplicit 3D-forståelse. Vi bruger Non-Rigid Structure-from-Motion (NRSfM) til at gendanne den underliggende 3D-struktur fra genererede 2D-billeder og foreslår en ny metode til at forbinde NRSfM-modellen med StyleGANs latente rum, hvilket muliggør eksplicit kontrol med 3D geometrien af de genererede billeder. For nylig er DDMs kommet frem som en stærk konkurrent til GANs, både hvad angår kvalitet og diversitet af de genererede billeder. Dog er DDMs' latente rum stadig ikke godt forstået. Den sidste del af denne afhandling foreslår nye metoder til semantisk redigering af ansigtsbilleder ved hjælp af DDMs.

Contents

Chapter 1

Introduction

1.1 Background and motivation

Deep Generative Models (DGMs) have seen impressive advancements in recent years, achieving unprecedented capabilities in generating photorealistic images across diverse domains. DGMs take a data-driven approach to image generation as these models aim to capture the underlying distributions of given data sets, enabling the generation of new images. Importantly, while the generated images are not contained in the training data itself, they follow the distribution of the training data. Thus, DGMs are able to generate entirely novel images from the domain of the training data. In particular, current state-of-the-art DGMs are able to generate near-perfect images of human faces. This PhD thesis focuses on the use of DGMs as a tool for modeling images of human faces and expressions and proposes several novel methods for controlling the output of DGMs in this context.

Before the advent of powerful DGMs, the primary method for generating synthetic facial images in computer graphics was based on creating 3D models and rendering textures onto them. This traditional method offers explicit and fine-grained control over the generated images: the 3D structure can be precisely manipulated by altering the underlying mesh, textures can be selectively applied, and lighting conditions and camera positions can also be controlled explicitly. However, despite the high level of control allowed by traditional computer graphics methods these

methods still fall short of achieving true photorealism when modeling human faces. As humans we have evolved a strong perceptual sensitivity to faces and can detect even very subtle deviations from realistic face images. Thus, even with a field as mature as 3D modeling, there arguably still remains an "uncanny valley" (Mori et al., 2012) when using traditional 3D modeling techniques to generate images of human faces.

On the other hand, modern DGM architectures like Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) and, more recently, Denoising Diffusion Models (DDMs) (Sohl-Dickstein et al., 2015) are able to synthesize images with a high level of photorealism. In particular, the StyleGAN (Karras et al., 2019, 2020, 2021) family of models has shown a remarkable capability in producing photorealistic face images that are almost indistinguishable from real images. For this reason StyleGAN, has been described as the de facto gold standard (Bermano et al., 2022) for face synthesis applications. In fact, a recent study by Tucciarelli et al. (2022) examined the ability of human subjects to differentiate between real photographs and face images generated by StyleGAN. Surprisingly, the results indicated that StyleGAN images were frequently perceived as more authentic than real images by human evaluators. However, despite the impressive photorealism of images produced by models like StyleGAN, modern DGMs typically fall short of the fine-grained control over facial features, expressions, and orientation that more traditional computer vision methods offer.

The central aim of this PhD thesis is to develop novel methods for attaining more explicit control over the images generated by DGMs, particularly in the context of synthesizing human portrait images. At a high level, DGMs are models that learn latent representations from their training data and organize the information into a *latent space*. This thesis aims to develop new methods to discover preferred directions or trajectories in the latent space of DGMs that encode semantic information that we might be interested in controlling, such as the ability to explicitly control the pose or facial expression of the generated face images.

1.2 Project description and research objectives

The primary objective of this PhD project is to advance the understanding of how DGMs can be used to model human faces, including emotions and facial expressions as well as the 3D geometry of the generated images. A central objective is to develop methods to discover meaningful subspaces and directions within the latent space of these models that have a clear semantic interpretation. The project explores how traditional facial modeling methods can be integrated with state-of-the-art DGMs in order to leverage the learned latent representations to achieve greater control over the images generated with DGMs.

In particular, this PhD thesis aims to answer the following research questions.

- How can we find preferred directions in the latent space of DGMs that change only a single, semantically meaningful, attribute of the generated images, such as the pose or facial expression?
- Can a multilinear treatment be used to factorize the latent space of DGMs into interesting subspaces that each control different semantic content of the generated image, for example the pose or facial expression?
- Modern DGMs can generate high-quality 2D images but have no explicit 3D understanding. Can we use traditional approaches for 3D reconstruction, such as Non-Rigid Structure-from-Motion (NRSfM), to get explicit control over the 3D structure of the images generated by modern DGMs?

1.3 Thesis contributions

This PhD thesis has made several contributions in the field of deep generative modeling with applications for semantic editing of face portrait images. In the following, I will summarize the main scientific contributions of this thesis. While the first four papers focus on StyleGAN, as the primary object of study, the last paper proposes novel editing techniques in the semantic latent space of DDMs.

First, in Paper I and Paper II, this thesis provides the first attempt at combining a multilinear tensor model with current state-of-the-art deep generative models like StyleGAN. In the papers, we showed that, using a real facial expression data set as supervision, multilinear tensor models can successfully uncover meaningful semantic subspaces in StyleGAN. Specifically we found subspaces corresponding to the six prototypical human emotions as well as a direction corresponding to yaw rotation. Coupled with a state-of-the-art technique for GAN inversion, our method proved an effective framework for editing the facial expressions of real face portrait images.

Second, in Paper III, this thesis provides the first approach for combining NRSfMs, which has a long-standing history in computer vision, with the latent space of deep generative models like StyleGAN. By connecting NRSfM with the latent space of StyleGAN, our method allows for attaining more explicit control over the 3D structure and camera orientation of face images generated with StyleGAN.

Third, in Paper IV, this thesis proposes an approach for using DGMs to disseminate large art collections that are too vast to adequately exhibit in a traditional museum setting. The work contributed with an art installation in the form of an interactive drawing table where the audience directly interacts with a trained StyleGAN model by providing pen and paper sketches as the input to the system. The system then translates the sketch provided by the user into an image in the domain of the training data – in this case, a large collection of hand-drawn sketches by the Norwegian painter Edvard Munch.

Finally, in Paper V, this thesis contributes to the emerging field of diffusion models by proposing novel supervised and unsupervised methods for discovering semantically meaningful directions in the semantic latent space of diffusion models. In particular, we are among the first to propose methods that facilitate semantic image editing using DDMs in an entirely unsupervised fashion.

List of papers.

The following list contains all papers which have been submitted or accepted for international conferences during my PhD. Each of these papers can be found in full in Chapter A.

- Paper I. **Tensor-based Subspace Factorization for StyleGAN**
  René Haas, Stella Graßhof and Sami Sebastian Brandt
  Published in the proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition 2021 (FG2021)
- Paper II. **Tensor-based Emotion Editing in the StyleGAN Latent Space**
  René Haas, Stella Graßhof and Sami Sebastian Brandt
  Accepted for presentation at the AI for content creation workshop at the IEEE / CVF Computer Vision and Pattern Recognition Conference 2022 (CVPR2022)
- Paper III. **Controllable GAN Synthesis Using Non-Rigid Structure-from-Motion**
  René Haas, Stella Graßhof and Sami Sebastian Brandt
  Accepted for presentation at the Generative Models for Computer Vision workshop at the IEEE / CVF Computer Vision and Pattern Recognition Conference 2023 (CVPR2023)
- Paper IV. **Exploring a Digital Art Collection through Drawing Interactions with a Deep Generative Model**
  Christian Sivertsen, René Haas, Halfdan Hauch Jensen and Anders Sundnes Løvlie
  Accepted for presentation at the CHI Conference on Human Factors in Computing Systems 2023 (CHI2023)
- Paper V. **Discovering Interpretable Directions in the Semantic Latent Space of Diffusion Models**
  René Haas, Inbar Huberman-Spiegelglas, Rotem Mulayoff, Stella Graßhof, Sami Sebastian Brandt and Tomer Michaeli
  Conference paper under review.

1.4 Thesis outline

The rest of this thesis is organized as follows.

- Chapter 2 presents a general overview of core concepts and applications of GANs. The chapter will begin with a general introduction to GANs, and continue with an introduction to the StyleGAN architecture. The chapter proceeds with a review of the work of other researchers focusing on the topics of GAN inversion and semantic editing.
- Chapter 3 introduces Paper I and Paper II. The chapter begins with an overview of core concepts and operations from multilinear algebra. It then continues with an introduction to our proposed method, that uses a multilinear model to factorize the latent space of StyleGAN into various semantically meaningful subspaces.
- Chapter 4 introduces Paper III. The chapter begins with an introduction of NRSfM problem and continues with a summary of our proposed method for combining NRSfM with the latent space of StyleGAN in order to achieve more explicit control of the 3D structure of face images generated by the model.
- Chapter 5 introduces Paper IV. The paper proposes using StyleGAN as a tool for creating an interactive art experience. Here the audience can interact directly with the model in order to explore the sketching style of the famous Norwegian painter Edvard Munch.
- Chapter 6 introduces Paper V. The chapter begins with a general introduction to DDMs and continues with an introduction to the recently proposed semantic latent space of DDMs and to our proposed supervised and unsupervised approaches for editing of face images using the semantic latent space.
- Chapter 7 begins with a discussion of the various ethical considerations that arise with the advent of powerful image generation systems, in particular in relation to DGMs capable of synthesizing photorealistic human faces. The chapter ends with a discussion of the limitations of the methods proposed in this thesis as well as perspectives for future work.

Chapter 2

StyleGAN: Architecture and applications

This chapter provides an introduction to GANs with special emphasis on the StyleGAN architecture. In recent years, GANs has emerged as one of the most successful approaches in generative modeling of images. StyleGAN, a state-of-the-art GANs architecture, has gained significant attention for its impressive performance in generating images with near-perfect photorealism. StyleGAN excels in the generation of images from structured domains such as images of human faces and the architecture is a core object of study of this thesis.

The chapter begins with a brief overview and history of GANs in Section 2.1. Section 2.2 will introduce different metrics that are commonly used to evaluate the performance of DGMs as well as to measure the similarity between images and, in the context of face images, evaluate the degree of identity similarity. Next, Section 2.3 introduces StyleGAN and gives a brief overview of its evolution along the three main versions of the architecture. Section 2.4 gives an introduction to different methods for GAN inversion. Finally, the chapter ends in Section 2.5 with an introduction of different methods for utilizing the latent space of StyleGAN for semantic editing of the generated images.

(a) **Evolution of GANs**

(b) **StyleGAN samples.**

Figure 2.1: **Evolution and quality of face images synthesized by GANs.** (a) Tweet by Ian Goodfellow illustrating the rapid improvements in the quality of GAN generated face images in recent years. (Goodfellow, 2019) (b) Modern StyleGAN models are able to synthesize images of human faces that are almost indistinguishable from real portrait photos. The images are created using StyleGAN2 (Karras et al., 2020).

2.1 Generative Adversarial Networks

GANs have gained widespread popularity in recent years. Since the original proposal by Goodfellow et al. (2014), there have been rapid improvements to the image quality of face images produced by various GANs architecture as illustrated in Figure 2.1a. Currently, the style-based generator architecture (StyleGAN) (Karras et al., 2019) is one of the most widely used and successful GAN architectures. A selection of samples from a StyleGAN2 (Karras et al., 2020) model trained on the Flickr-Faces-HQ (FFHQ) (Karras et al., 2019) data set is shown in Figure 2.1b. The figure demonstrates the impressive face generation capabilities of StyleGAN models as it is not a trivial task for a human to identify that these images are not real photographs.

The basic premise of GANs is conceptually simple and on a high level it can be formulated as a game between two competing neural networks: a generator and a discriminator. The generator's objective is to create images that are realistic and resemble the images in the training data set. The input to the generator is a ran-

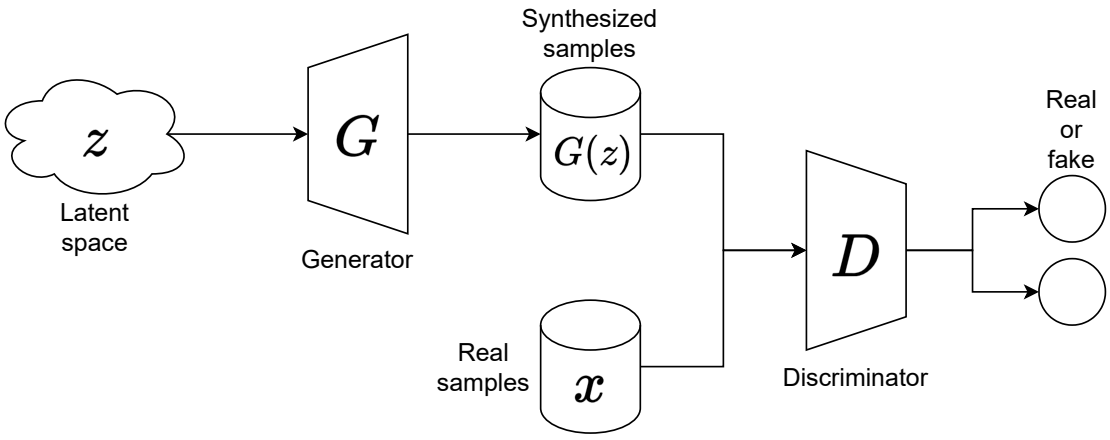


Figure 2.2: **High-level architecture of a Generative Adversarial Networks.** GANs train to networks simultaneously. The generator synthesizes new samples resembling the training data and the discriminator is tasked to differentiate between the real and generated images.

dom noise vector $\mathbf{z}$ that is drawn some prior distribution $p_{\mathbf{z}}$. Typically $p_{\mathbf{z}}$ is chosen to be the standard multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The generator then transforms the random noise vector into an image in a deterministic way such that a particular realization of the random noise will correspond to a particular image. The discriminator is a binary classifier whose objective is to classify whether a given image is coming from the data distribution (*i.e.*, the training data set) or the generated distribution (*i.e.*, produced by the generator). The generator and discriminator are trained simultaneously, with the generator attempting to produce more realistic images and the discriminator becoming better at distinguishing between real and fake images. This iterative process continues until the discriminator is no longer able to distinguish the generated images from the training data. A diagram of the GAN training setup is shown in Figure 2.2.

In the original paper by Goodfellow et al. (2014), both the discriminator and generator architectures were parameterized as simple Multilayer Perceptrons (MLPs). The original training objective was formulated as

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]. \tag{2.1}$$

During training, the weights of the generator and discriminator are updated in

an alternating fashion. Goodfellow et al. (2014) proposed to alternating between updating D for k-steps and updating G for a single step in order to ensure that D is kept near its optimal solution during training.

To update the weights of the discriminator a set of m noise samples $\{\mathbf{z}_1, \cdots, \mathbf{z}_m\}$ are drawn from the noise prior $p_{\mathbf{z}}$ as well as a set of real images $\{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$ from the data distribution p_{data}. The weights of the discriminator θ_d can then be updated by ascending the gradient

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D(\mathbf{x}_i) + \log \left(1 - D(G(\mathbf{z}_i)) \right) \right]. \tag{2.2}$$

Correspondingly the weights of the generator θ_g are updated by descending the gradient

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \left[\log \left(1 - D(G(\mathbf{z}_i)) \right) \right]. \tag{2.3}$$

In the paper, Goodfellow et al. notes that the term $\log(1 - D(G(\mathbf{z})))$ in Eq. (2.1) has a small gradient early in training when G is still bad (*i.e.*, when the probability of images being classified as fake is still high). Instead Goodfellow et al. proposes a *non-saturating loss* that trains G to maximize $\log D(G(\mathbf{z}))$ (or equivalently minimize $-\log D(G(\mathbf{z}))$). Intuitively this corresponds to a shift in perspective where G tries to maximize the probability of images being classified as real rather than minimizing the probability of the images being classified as fake. This small change gives stronger gradients early in training while it does not affect the fixed point dynamics of either G or D.

To gain insights into the optimally conditions of the min-max objective, Eq.(2.1) can be expanded into its integral form

$$\min_G \max_D V(D, G) = \int_{\mathbf{x}} \left[p_{\text{data}}(\mathbf{x}) \log(D(\mathbf{x})) + p_G(\mathbf{x}) \log(1 - D(\mathbf{x})) \right] d\mathbf{x}, \tag{2.4}$$

where a change of variables has been used to write

$$\int_{\mathbf{z}} p_{\mathbf{z}}(\mathbf{z}) \log(1 - D(G(\mathbf{z}))) d\mathbf{z} = \int_{\mathbf{x}} p_G(\mathbf{x}) \log(1 - D(\mathbf{x})) d\mathbf{x}. \tag{2.5}$$

The integrand in Eq.(2.4) has the form $a\log(y)+b\log(1-y)$ which has an extremum at $y=\frac{a}{a+b}$ for $y\in[0,1]$. Thus for a fixed generator G, the optimal discriminator D^* can be written in terms of the data and generator distributions as

$$D^*(\mathbf{x})=\frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x})+p_G(\mathbf{x})}. \tag{2.6}$$

Before proceeding, recall the definition of the Kullback–Leibler divergence

$$D_{\text{KL}}(P||Q)=\mathbb{E}_{x\sim P(x)}\left[\log\frac{P(x)}{Q(x)}\right] \tag{2.7}$$

and the Jensen-Shannon divergence

$$D_{\text{JS}}(P||Q)=\frac{1}{2}D_{\text{KL}}\left(P\middle|\middle|\frac{P+Q}{2}\right)+\frac{1}{2}D_{\text{KL}}\left(Q\middle|\middle|\frac{P+Q}{2}\right). \tag{2.8}$$

The JS divergence goes to zero if the distributions match and unlike the KL-divergence the JS divergence is symmetric, *i.e.*, $D_{\text{JS}}(P||Q)=D_{\text{JS}}(Q||P)$.

We can plug the expression of the optimal discriminator in Eq. (2.6) into the cost function in Eq. (2.1) to get an upper bound on the loss

$$\begin{aligned}V(D^*,G)&=\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\left[\log\frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x})+p_G(\mathbf{x})}\right]+\mathbb{E}_{\mathbf{x}\sim p_G(\mathbf{z})}\left[\log\frac{p_G(\mathbf{x})}{p_{\text{data}}(\mathbf{x})+p_G(\mathbf{x})}\right]\\&=-2\log 2+\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\left[\log\frac{p_{\text{data}}(\mathbf{x})}{\frac{1}{2}(p_{\text{data}}(\mathbf{x})+p_G(\mathbf{x}))}\right]\\&\qquad+\mathbb{E}_{\mathbf{x}\sim p_G(\mathbf{z})}\left[\log\frac{p_G(\mathbf{x})}{\frac{1}{2}(p_{\text{data}}(\mathbf{x})+p_G(\mathbf{x}))}\right]\\&=-\log 4+2D_{\text{JS}}(p_{\text{data}}||p_G).\end{aligned} \tag{2.9}$$

So for the optimal discriminator, the generator is trying to minimize the Jensen-Shannon divergence between the data distribution p_{data} and the generator distribution p_G. Eq. (2.9) has global minimum of $V(D^*,G)=-\log 4$ when the generator distribution perfectly matches the data distribution, *i.e.*, $p_G=p_{\text{data}}$. At this point the $D^*=1/2$ everywhere and the discriminator will be unable to differentiate between the two distributions.

Training GANs using the min-max objective in Eq. (2.1) can be difficult due to

issues such as mode collapse and vanishing gradients. Arjovsky et al. (2017) proposed Wasserstein GAN (WGAN) that makes adaptations to the training procedure of GANs in order to overcome these issues and make training more stable. The primary idea in WGAN is to replace the Jenson-Shannon divergence with the Wasserstein distance. In the min-max objective in Eq.(2.1) the discriminator D acts like a binary classifier while in WGAN the discriminator is utilized to approximate the Wasserstein distance, which is a regression task. For this reason the sigmoid activation in the last layer of D is removed so D is not constrained to output a number between zero and one. Since the discriminator is not trained as a classifier it is referred to as a "critic" in the paper by Arjovsky et al. (2017).

Using concepts from convex optimization (Villani et al., 2009), Arjovsky et al. (2017) proposed an implementation of the Wasserstein objective that requires D to be a 1-Lipschitz scalar function, *i.e.*, a function that obeys $|f(x)-f(y)| \leq |x-y|$. As a way to enforce the Lipschitz constraint, Arjovsky et al. (2017) proposed to clip the weights of D at every training iteration. With regard to the choice of using weight clipping is a simple way to enforce the Lipschitz constraint in D the authors note that

> "Weight clipping is a clearly terrible way to enforce a Lipschitz constraint. If the clipping parameter is large, then it can take a long time for any weights to reach their limit, thereby making it harder to train the critic till optimality. If the clipping is small, this can easily lead to vanishing gradients [...] we stuck with weight clipping due to its simplicity and already good performance." (Arjovsky et al., 2017, p.7)

Rather than relying on weight clipping, Gulrajani et al. (2017) proposed adding a "gradient penalty" term to the WGAN loss as an alternative way to enforce the Lipschitz constraint. This leads to the WGAN-GP training objective $\mathcal{L}_{\text{WGAN-GP}}$, than can be written as

$$\mathcal{L}_{\text{WGAN-GP}} = \underbrace{\mathbb{E}_{\mathbf{x}\sim p_G}[D(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[D(\mathbf{x})]}_{\text{Original WGAN loss}} + \lambda \underbrace{\mathbb{E}_{\mathbf{x}\sim \tau}[(||\nabla_{\mathbf{x}} D(\mathbf{x})||_2 - 1)^2]}_{\text{Gradient penalty term}}, \quad (2.10)$$

where λ is a strength parameter that is set to ten in the paper and $\tau := \tau(p_G, p_{\text{data}})$

is a distribution created from sampling uniformly along straight lines between pairs of points, sampled from the generator and data distributions respectively. The WGAN-GP objective in Eq. (2.10) leads to more stable training than the original GAN loss in Eq. (2.1).

DCGAN (Radford et al., 2016) is a GAN architecture which uses Convolutional Neural Networks (CNNs) as the architecture in both the generator and discriminator rather than the MLPs proposed by Goodfellow et al. (2014). In their paper, Radford et al. demonstrated the linear vector space property of the learned latent space of GANs for the first time. Treating the latent representations as semantic vectors where semantic manipulations can be formed as simple linear arithmetic operations was first shown in the context of word representations by Mikolov et al. (2013) who showed that using their word representations, the nearest neighbors of the resultant vector for *"King"* - *"Man"* + *"Woman"* is the vector representation of the word *"Queen"*. Radford et al. showed that the latent space of GANs has a similar property where semantic concepts in face images such as whether the person is smiling or has glasses can be edited using simple arithmetic operations in the latent space.

Progressive GAN (ProGAN) (Karras et al., 2018) is the immediate predecessor to StyleGAN which will be introduced in Section 2.3. The main idea of ProGAN is that rather than generating full-resolution images at the beginning of training, instead, the training takes place in several stages where the output resolution of the images is progressively increased. At each stage extra layers are added to both the generator and discriminator in order to accommodate the higher resolutions. ProGAN is the first GAN architecture that can produce near-photorealistic images with resolutions as high as 1024×1024.

2.2 Image Similarity Metrics

Before introducing StyleGAN in the next section, this section will introduce different metrics that have been used to measure the similarity of images, the similarity of faces in particular, and the similarity between image and text pairs. These metrics have been widely used both as cost functions for training and evaluating

GANs and as components in different approaches for semantic editing which will be discussed in Section 2.5.

A simple way to compare two images is to compare the pixel values directly. The L_2 distance quantifies the pixel-wise similarity of images and is defined as

$$L_2(\mathbf{x}, \mathbf{x}_0) = \|\mathbf{x} - \mathbf{x}_0\|_2 . \tag{2.11}$$

There are several reasons why the L_2 distance between two images is not always a good similarity metric. For example, if an image is translated a few pixels in any one direction, the resultant image could potentially have a very large L_2 distance when compared to the original, even though humans would have no problem identifying the two images as being essentially the same.

The Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018) metric is widely used to measure the perceptual similarity between pairs of images and has been specifically designed to capture the way that humans perceive image similarity. The LPIPS between two images $\mathbf{x}$, $\mathbf{x}_0$ is calculated by first feeding them to a feature extraction network $\mathcal{F}$. Here, the network $\mathcal{F}$ is some CNN where the exact choice of architecture is flexible. In the paper, Zhang et al. showed that LPIPS can be implemented using either a VGG (Simonyan and Zisserman, 2015) network, AlexNet (Krizhevsky et al., 2012) or SqueezeNet (Iandola et al., 2016). Regardless of architecture, features are extracted using network $\mathcal{F}$ from L layers and by denoting the normalized feature map activation at layer l for the image as $\mathbf{y}^l, \mathbf{y}_0^l \in \mathbb{R}^{H_l, W_l, C_l}$, the LPIPS between the images $\mathbf{x}$ and $\mathbf{x}_0$ can be calculated as

$$\text{LPIPS}(\mathbf{x}, \mathbf{x}_0) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \left\| \mathbf{w}^l \odot (\mathbf{y}_{hw}^l - \mathbf{y}_{0,hw}^l) \right\|_2^2 , \tag{2.12}$$

where $\mathbf{w}^l \in \mathbb{R}^{C_l}$ is a learnable weight that scales the channel dimensions as each layer. Zhang et al. (2018) show that LPIPS often agrees more with the similarity judgements made by humans than more traditional similarity metrics.

Arcface (Deng et al., 2019) is a similarity metric that takes two images as input and outputs a measure of how likely it is that the two images are of the same person. Measuring identity similarity is a challenging task since the system needs to

be robust to variations in pose, expression, illumination, *etc.*, and still recognize if the images are of the same person. Arcface achieves good performance in this task by replacing the softmax loss, which is typically used for multi-class classification problems, with a proposed Angular Margin Loss which ensures that the angle between feature embedding is maximized for different classes *i.e.* different identities. In the context of analyzing DGMs in the domain of human faces, Arcface is often used to define an *identity loss* as

$$\mathcal{L}_{ID}(\mathbf{x}_0, \mathbf{x}) = 1 - \langle \mathcal{F}(\mathbf{x}), \mathcal{F}(\mathbf{x}_0) \rangle, \tag{2.13}$$

where $\mathcal{F}(\mathbf{x})$ denotes features extracted with the Arcface network $\mathcal{F}$. In Paper III, we use Arcface as regularization term in order to increase the degree of identity preservation when performing semantic edits of face images.

The Fréchet Inception Distance (FID) (Heusel et al., 2017) has become the standard metric evaluating sample quality and diversity in DGMs. FID calculates the Fréchet Distance (Dowson and Landau, 1982) between two Gaussian Distributions $\mathcal{N}(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$ and $\mathcal{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ where the feature representations of the images are calculated using a pretrained and frozen Inception network (Szegedy et al., 2016). The FID score can be calculated as

$$\text{FID} = \|\boldsymbol{\mu}_r - \boldsymbol{\mu}_g\| + \text{Tr}(\boldsymbol{\Sigma}_r + \boldsymbol{\Sigma}_g - 2(\boldsymbol{\Sigma}_g \boldsymbol{\Sigma}_r)^{1/2}). \tag{2.14}$$

Contrastive Language-Image Pretraining (CLIP) (Radford et al., 2021) models offer a way to measure the similarity between text and images pairs. Radford et al. (2021) trained their model on a large data set consisting of 400 million image and text pairs which are collected from the internet. The idea is to extract representations from both the images and text and project each into a shared embedding space. In this space, the similarity can be measured by calculating the cosine similarity between the text and image representations. The text encoder in CLIP is a transformer (Vaswani et al., 2017) architecture and in the paper, Radford et al. (2021) experimented with both ResNet (He et al., 2016) and Vision Transformer (ViT) (Dosovitskiy et al., 2021) architectures for the image encoder where the ViT model was found to perform the best. CLIP allows for easy zero-shot image classification simply by creating a textual description for each of the desired classes

Figure 2.3: **Interpolations in latent space** The latent space of StyleGAN is smooth and highly disentangled, allowing for smooth interpolations between pairs of images.

and comparing the extracted image features to the textual features corresponding to each class. In Paper V we use the zero-shot classification capability of CLIP to evaluate the effectiveness of our proposed method for disentangling semantic directions in the semantic latent space of DDMs.

2.3 StyleGAN

This section will outline the theory, evolution, and applications of StyleGAN (Karras et al., 2019). StyleGAN offers impressive performance in image generation and can synthesize images with a near-perfect photorealism. Since its initial release, StyleGAN has emerged to become one of the most well-studied generative models in recent years, and has been called a de facto gold standard for the synthesis and editing of face images (Alaluf et al., 2023). One of the core strengths of StyleGAN models is their smooth and highly disentangled latent space. This property allows for linear interpolations between different pairs of images where each intermediate image is meaningful by itself, and where the image features vary smoothly along the interpolation path. Figure 2.3 shows examples of interpolations in the latent space of StyleGAN between three pairs of images.

The StyleGAN architecture draws inspiration from the literature on style transfer (Huang and Belongie, 2017). In a traditional GAN architecture, a latent code is initially sampled from a known prior distribution and then fed to the generator through an input layer, *i.e.*, the first layer in a feed-forward or convolutional neural network.

There are two main ways in which the StyleGAN generator differs from that of a traditional GAN architecture. First, the introduction of an intermediate learned latent space, and secondly, the way in which latent codes influence the synthesis process. The StyleGAN *generator* $G : \mathcal{Z} \to \mathcal{X}$ is composed of two networks, a *mapping network* $f : \mathcal{Z} \to \mathcal{W}$ and a *synthesis network* $g : \mathcal{W} \to \mathcal{X}$ that outputs the generated image $\mathbf{x} \in \mathcal{X}$. The following will account for how the introduction of the mapping network encourages a more disentangled latent space before continuing to describe how the latent codes influence the image generation process in the synthesis network.

The mapping network f maps latent codes $\mathbf{z} \in \mathcal{Z}$, from the Gaussian latent space $\mathcal{Z}$ onto an intermediate latent space $\mathcal{W}$. The motivation for the inclusion of such an intermediate mapping network is to encourage disentanglement of the latent space. We say that a latent space is entangled if we, when interpolating between two points in latent space, observe features along the interpolation trajectory that are absent in both endpoints. For example, consider an interpolation between two images depicting a man and a woman, both without glasses. If glasses suddenly appear along the interpolation trajectory that would signify an entangled latent space. As an example of why entanglement may arise, consider a simplified representation where all human faces can be represented on a two-dimensional plane. For the sake of argument, assume that the two available factors of variation correspond to the attributes of gender and facial hair. When sampling from the Gaussian $\mathcal{Z}$ space, the probability of each combination of factors must match that of the training data. However, data sets might not have an equal representation of all possible factors of variation. In the current example, we may imagine that women with facial hair are absent in the training data. This absence of a specific combination of factors causes the latent space to become warped such that the invalid combinations disappear in the latent space and consequently it becomes entangled (Karras et al., 2019).

The argument made by Karras et al. (2019) for the introduction of the mapping network, is that it is able to "undo" much of the warping since the intermediate latent space does not have follow any fixed distribution. Karras et al. further notes that there is as natural pressure for the generator to learn disentangled representations in the intermediate latent space since it should be easier for the generator to generate realistic images based on a more disentangled representation rather than an entangled one.

Perceptual Path Length (PPL) was introduced in the original StyleGAN paper (Karras et al., 2019), as a way quantify the degree of disentanglement of the latent space of the trained generator. In $\mathcal{W}$ space[1] the PPL metric can be written in closed form as

$$l_{\mathcal{W}} = \mathbb{E}\left[\frac{1}{\epsilon^2} d(g(\text{lerp}(\mathbf{w}_1, \mathbf{w}_2; t)), g(\text{lerp}(\mathbf{w}_1, \mathbf{w}_2; t + \epsilon)))\right], \tag{2.15}$$

where $\mathbf{w}_1, \mathbf{w}_2 \sim \mathcal{W}$, ϵ denotes a small subdivision that the authors set to 10^{-4}, lerp denotes linear interpolation, *i.e.*, $\text{lerp}(\mathbf{w}_1, \mathbf{w}_2, t) = (1-t)\mathbf{w}_1 + t\mathbf{w}_2$, t is sampled from the uniform distribution $U(0, 1)$ and $d(\cdot, \cdot)$ measures perceptual similarity between the two generated images, *i.e.*, the distance measure d can be implemented as LPIPS. Using the PPL metric, Karras et al. (2019) showed that the intermediate $\mathcal{W}$ space is quantitatively more disentangled than the $\mathcal{Z}$ space.

Figure 2.4 provides a qualitative illustration highlighting the superior degree of disentanglement in $\mathcal{W}$ space as compared to $\mathcal{Z}$ space. As seen in Figure 2.4b, interpolations in $\mathcal{W}$ space lead to a smooth transition between the endpoint images, where the middle images exhibit a balanced combination of attributes from both endpoints. Conversely, interpolating between the same two images in $\mathcal{Z}$ space results in exaggerated attributes in the interpolated images, which do not form a balanced mixture of the endpoints' characteristics. For example, we observe significantly larger eyeglasses and a darker skin tone in the interpolated images than in either of the two endpoints.

Unlike a traditional GAN architecture where image synthesis begins from the latent

[1] To calculate the PPL distance in $\mathcal{Z}$ space the synthesis network g should be replaced by the full generator $G = g \circ f$ and spherical linear interpolation (slerp) should be used rather than linear interpolation (lerp) due to the normalization of the input latent codes in $\mathcal{Z}$ space.

(a) Interpolation in $\mathcal{Z}$ space. (b) Interpolation in $\mathcal{W}$ space.

Figure 2.4: **The mapping network encourages a disentangled latent space.** Karras et al. (2019) found that the addition of a mapping network $f : \mathcal{Z} \to \mathcal{W}$ to the GAN architecture allows the model to learn an intermediate latent space $\mathcal{W}$ that is more disentangled than the Gaussian $\mathcal{Z}$ space. By comparing interpolations in $\mathcal{Z}$ and $\mathcal{W}$, we clearly see that the intermediate $\mathcal{W}$ space is more disentangled and offers a smoother transition between the two endpoints of the interpolations.

code, StyleGAN initiates the synthesis process from a learned constant $4 \times 4 \times 512$ tensor. The synthesis network consists of a series of synthesis blocks, each containing two layers. After the first block, a 2x upsampling operation occurs at the beginning of each subsequent block. Thus, the full 1024×1024 resolution synthesis network has 9 blocks and a total of 18 synthesis layers. The mapping network produces an intermediate latent code $\mathbf{w} \in \mathcal{W}$. This latent code is copied and then separately fed to each affine transformation associated with each of the synthesis layers. The output from each affine transformation is then used to influence the synthesis process. The exact process by which the output of the affine transformations influence the synthesis has evolved through the three main versions of the StyleGAN architecture.

The first version of StyleGAN (Karras et al., 2019) used the progressive growing approach proposed by Karras et al. (2018) to gradually increase the resolution of the generated images during training. After the initial mapping to the intermediate latent space $\mathbf{w} \in \mathcal{W}$, the latent code $\mathbf{w}$ is transformed into style codes $\mathbf{y} = (\mathbf{y}_s, \mathbf{y}_b)$ using learned affine transformations (see Figure 2.5), which then act on the intermediate feature maps throughout the synthesis network g using the Adaptive Instance Normalization (AdaIN) operation. The AdaIN operation can be written as

$$\text{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i}, \tag{2.16}$$

where each feature map is first normalized and then scaled and biased accord-

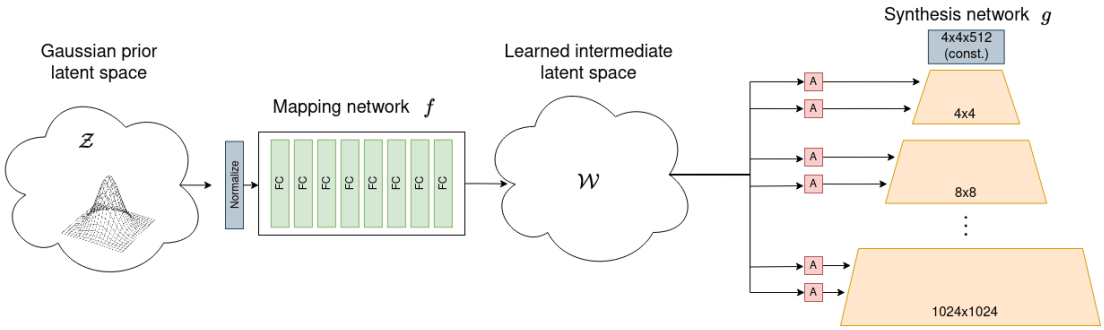


Figure 2.5: **Illustration of the StyleGAN architecture.** The StyleGAN mapping network first transforms latent codes from a Guassian latent space $\mathcal{Z}$ to an intermediate learned latent space $\mathcal{W}$. The latent code $\mathbf{w} \in \mathcal{W}$ is then copied and passed to a set of learned affine transformations ("A" in the figure) which influence the synthesis process using either the AdaIN operation in StyleGAN1 (Karras et al., 2019) or by a "demodulation" operation which is applied to the convolutional weights in StyleGAN2 (Karras et al., 2020).

ing to the style vector $\mathbf{y}$. Thus, in the first incarnation of StyleGAN, the latent codes influence the synthesis process by directly affecting the feature maps through AdaIN.

Although StyleGAN1 is able to synthesize highly realistic and high-resolution images, the architecture had some flaws that adversely affected the quality of the generated images. The subsequent paper, StyleGAN2 (Karras et al., 2020), focused on alleviating these limitations and improving the architecture.

While the progressive growing methodology introduced by Karras et al. (2018) is effective in stabilizing the training of high-resolution generators, it also introduced its own problems. Karras et al. (2020) noted that progressive growing leads to "phase artifacts" where certain features like the teeth or eyes have a strong location preference. This means that these features would stay in one place before quickly jumping to the next preferred location instead of varying smoothly when editing pose. In StyleGAN2 (Karras et al., 2020), this problem was solved by removing the progressive growing of the generator and instead incorporating skip connections into the generator and increasing the number of feature maps in the layers of the generator corresponding to higher resolutions.

Another problem with the original architecture is that all images synthesized by

StyleGAN1 had characteristic water droplet-like artifacts. Although these were not always immediately visible in the generated images, they would always be present in the feature maps of the generator after a certain resolution Karras et al. (2020). The authors pinpointed the problem to the use of AdaIN as the course of these blob-like artifacts. As an alternative to the AdaIN operation, StyleGAN2 introduces convolution modulation and demodulation operations that acts directly on the convolutional weights as

$$w''_{ijk} = w'_{ijk} / \sqrt{\sum_{i,k} w'^2_{ijk} + \epsilon} \quad \text{with} \quad w'_{ijk} = s_i w_{ijk}, \tag{2.17}$$

where w_{ijk} denotes the convolution weights (as opposed $\mathbf{w}$ that denotes a latent code). The insight here is that, instead of controlling the synthesis process with instance normalization applied to the feature maps, the same control can be achieved by modulating and demodulating the convolution kernel weights directly.

The third version of StyleGAN (Karras et al., 2021) addressed another limitation present in the previous two architectures. The authors observed a "texture sticking" phenomenon in StyleGAN1 and StyleGAN2 where, when interpolating in latent space, fine details like the hair would not move naturally along with the location of the face, but rather appear to be stuck to certain pixel locations. StyleGAN3 is specifically designed to overcome this texture sticking problem and at the same time the authors propose changes to the architecture that make the generator equivariant to both translation and rotation. Along with several other architectural changes, StyleGAN3 replaces the learned $4 \times 4 \times 512$ constant input tensor with Fourier Features, allowing for continuous translations and rotations to be applied to the input. This endows the StyleGAN3 generator with explicit control over translation and rotation.

In the original StyleGAN paper (Karras et al., 2019), the authors proposed to use a *truncation trick* to gain explicit control over a trade-off between the quality and diversity of generated samples. To employ the truncation trick, the center of mass is first computed in $\mathcal{W}$ space as

$$\bar{\mathbf{w}} = \mathbb{E}_{\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[f(\mathbf{z}) \right], \tag{2.18}$$

where $\bar{\mathbf{w}}$, in the case of a model trained on FFHQ, can be interpreted as the "average face" of the data set. Using the truncation trick, a truncated latent code, $\mathbf{w}'$ can then be written as a linear combination of a sampled latent code $\mathbf{w}$ and the center of mass $\bar{\mathbf{w}}$ using the truncation parameter ψ with $0 \leq \psi < 1$ as

$$\mathbf{w}' = \bar{\mathbf{w}} + \psi(\mathbf{w} - \bar{\mathbf{w}}). \tag{2.19}$$

Lower values of the truncation parameter ψ steer the generated images closer to the mean of the data distribution, which improves image quality but reduces image diversity. On the contrary, higher values of ψ produce images that are more diverse but generally of lower quality with a higher rate of artifacts.

While traditional GAN models have a singular well-defined latent space, StyleGAN has several innate spaces that can be considered as the latent space. These different spaces correspond to different stages in the synthesis process and offer different strengths and weaknesses depending on the application. We have already touched on the Gaussian $\mathcal{Z}$ space and the intermediate, learned latent space $\mathcal{W}$. In the following sections on GAN inversion and latent space editing, we will see two additional spaces that have been proposed in the literature, denoted as $\mathcal{W}+$ and $\mathcal{S}$ space respectively.

2.4 GAN Inversion

Contrary to Variational Autoencoders (VAEs) (Kingma and Welling, 2014), GANs do not have an encoder as part of their design. For any application involving the editing of real images, it is necessary to first find a good latent representation for the target image. This problem is known as *GAN inversion* and was first introduced by Zhu et al. (2016). Specifically, we seek to find a latent code that, when passed to the generator, both faithfully reconstructs the target image and also resides in a region of the latent space that has properties suitable for semantic editing.

There are three main lines of techniques for GAN inversion in the context of StyleGAN. These techniques are either (1) optimization-based, (2) encoder-based, or more recently (3) methods that fine-tune the StyleGAN generator in order to

reconstruct a given target image. Traditionally there has been a trade-off between reconstruction quality vs. inference time when comparing optimization and encoder-based approaches. While optimization-based approaches tend to give higher-quality reconstructions, they are also slow, typically taking several minutes to invert a single image. On the other hand, encoder-based techniques are fast, requiring only a single forward pass through a trained encoder network, however typically as the cost of a lower reconstruction quality. As we will see, combining encoder-based approaches with techniques such as iterative refinement and fine-tuning of the generator has largely solved these issues allowing for both fast and accurate reconstructions.

Early work (Abdal et al., 2019; Nikitko., 2019) used an optimization-based approach where the latent code $\mathbf{w}$ is iteratively optimized with gradient decent such that the distance between the generated images $G(\mathbf{w})$ is as close as possible to some target image I. These optimization-based approaches can be formulated in terms of a minimization objective that minimizes the LPIPS and L_2 distance between the generated image $G(\mathbf{w})$ and target image I as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \text{LPIPS}(G(\mathbf{w}), I) + \lambda \left\| G(\mathbf{w}) - I \right\|_2^2 . \tag{2.20}$$

Abdal et al. (2019) proposed to optimize over latent codes in an extension of the intermediate latent space denoted as $\mathcal{W}+$ space. The $\mathcal{W}+$ space is the space arising from allowing the latent code to be different for each layer of the generator. Thus, while $\mathcal{W}$ is a 512-dimensional space, the $\mathcal{W}+$ space consists of a collection of different 512-dimensional latent codes corresponding to each layer. As mentioned earlier, the full 1024×1024 resolution generator has 18 layers, so a latent code in $\mathcal{W}+$ space has $18 \times 512 = 9216$ dimensions. Abdal et al. (2019) found that inversion into the extended $\mathcal{W}+$ space leads to considerably lower reconstruction error than inversion into the native $\mathcal{W}$ space. In the StyleGAN2 paper, Karras et al. (2020) pointed out that although inversion into the $\mathcal{W}+$ enables a projection that finds a closer match to the target image, it also enables the projection of arbitrary images that should not have a latent representation in a given trained model.

We use the optimization-based approach in Eq. (2.20) in Paper I in order to invert a facial expression data set of real images (Yin et al., 2006) into the $\mathcal{W}+$ space

of StyleGAN. Although this approach gives accurate reconstructions of the target images, latent codes found using this method are not necessarily ideal for editing applications as will be discussed later in this section.

Richardson et al. (2021) proposed the encoder-based pixel2style2pixel (pSp) framework which directly embeds an image into the extended $\mathcal{W}+$ space without requiring additional optimization. The pSp encoder works by first extracting feature maps from the input image using a feature pyramid (Dollár et al., 2014) over a ResNet (He et al., 2016) backbone. This creates three levels of feature maps, which are then used to extract latent codes that are fed to the StyleGAN generator before the learned affine transformations. The pSp architecture naturally facilitates a range of image-to-image translation tasks such as image inpainting, super-resolution, and face frontalization as well as conditioning the synthesis on a semantic mask or a user provided sketch. In Paper V, we use the image-to-image translation capability of the pSp encoder to create an interactive art experience where user-provided pen-and-paper sketches are transformed into images following the style of the Norwegian painter Edward Munch.

Alaluf et al. (2021) presented another approach that aims to close the gap between the usually slow but accurate optimization-based approaches and the faster but less accurate encoder-based methods. The approach also uses an encoder, but rather than asking the encoder to predict the latent code based on a single pass through the encoder network, Alaluf et al. (2021) instead proposes to supply the encoder with a current estimate, and let the encoder predict the residual with respect to the current estimate. That is, the encoder predicts how the current estimate of the latent code should be changed in order to improve the reconstruction. This allows for passing the current estimate to the encoder several times, thus allowing the encoder to refine its prediction in each pass. The authors denoted this process of repeated application of the encoder as *iterative refinement* and their method enables a significant speedup compared to optimization-based approaches while maintaining a good reconstruction quality.

While the $\mathcal{W}+$ space generally offers good results with respect to reconstruction quality, Tov et al. (2021) noted that the $\mathcal{W}+$ space is not necessarily the best choice of latent space if we also wish to edit the projected images by traversing the

latent space. Tov et al. suggested that there exists a trade-off between distortion and editability when selecting which latent space to project a given target image into. To overcome this issue Tov et al. proposed an encoder architecture that is specifically designed to project real images into "well-behaved" and editable regions of the latent space. The encoder builds on the architecture from the pSp framework proposed by Richardson et al. and adds two main approaches for encouraging the latent codes that are projected into the $\mathcal{W}+$ to lie as close to the native $\mathcal{W}$ space as possible. The first idea is to initially predict a single latent code $\mathbf{w} \in \mathcal{W}$ which is then extended to $\mathcal{W}+$ by learning a series of offset Δ_i from $\mathbf{w}$ such that the final predicted latent code can be written as $(\mathbf{w}, \mathbf{w} + \Delta_1, \mathbf{w} + \Delta_2, \cdots \mathbf{w} + \Delta_{N-1}) \in \mathcal{W}+$. This allows the network to first learn a coarse reconstruction, which is then refined by sequentially learning the offsets Δ_i. To enforce that the latent codes have high proximity to $\mathcal{W}$ a regularization loss $\mathcal{L}_{\text{d-reg}} = \sum_{i=1}^{N-1} \|\Delta_i\|_2$ is employed during the training of the encoder. Secondly, to further ensure that the individual style codes of the projected latent codes lie within the actual distribution of the $\mathcal{W}$ space, Tov et al. proposed to regularize the encoder by adding the prediction of a latent discriminator to the loss during training. The latent discriminator is trained to discriminate between two types of samples: real samples from the $\mathcal{W}$ space, which can be generated by passing Gaussian samples through the mapping network, and the latent codes that are learned by the encoder.

Although the inversion method proposed by Tov et al. (2021) provides a good trade-off between reconstruction quality and editability of the resultant latent codes, the reconstructed images are still noticeably different from the original target image. Another line of research seeks to overcome this by fine-tuning the generator to accommodate a near-perfect reconstruction of a given target image while also preserving editability. Roich et al. (2021) showed that real images can be projected into $\mathcal{W}$ space with a near-perfect reconstruction quality by fine-tuning the trained generator around the target image, thus circumventing the need for projecting into $\mathcal{W}+$ space.

Nitzan et al. (2022) proposed to use StyleGAN as a personalized face prior, focusing on modeling the key facial features of a particular person. The method works by first projecting a collection of real images of the same person into the latent space

using the pivotal tuning approach proposed by Roich et al. (2021). This gives a collection of "anchor" latent codes. Nitzan et al. then proposed to define the personalized space, denoted as $\mathcal{P}$ space, as the convex hull of the projected latent corresponding to the same person. The personalized face prior can then be used for applications such as super-resolution, inpainting, or semantic editing, while preserving the key facial characteristic of the person in question.

Alaluf et al. (2022) proposed the HyperStyle encoder which combines the ideas of pivotal tuning proposed by Roich et al. (2021) and the iterative refinement method proposed by Alaluf et al. (2021). The aim is to train a hyper-network that is tasked with predicting how the weights of the pretrained generator should be changed so as to best reconstruct the target image. Like the method proposed by Roich et al. (2021), HyperStyle starts out by estimating an initial latent code $\mathbf{w}_{\text{init}} \in \mathcal{W}$ that offers an approximate reconstruction using the original weights θ of the generators. Based on the initial reconstruction $g(\mathbf{w}_{\text{init}}, \theta)$, the hyper-network then predicts a set of weight offsets Δ_i which are used to update the weights of the generator according to $\hat{\theta}_i = \theta_i(1 + \Delta_i)$ such that the final reconstruction $g(\mathbf{w}_{\text{init}}, \hat{\theta})$ more closely matches the target image. By modifying the weights of the generator, rather than projecting into the extended $\mathcal{W}+$ space, HyperStyle sidesteps the distortion-editability trade-off altogether by only projecting images into the more editable $\mathcal{W}$ space while maintaining a good reconstruction quality. Further, HyperStyle offers a substantial speedup when compared to the per-image optimization procedure proposed by Roich et al. (2021). The latter could take up to a few minutes to fine-tune the generator weights for a single image. In contrast, inference with HyperStyle typically takes only approximately a second. Additionally, HyperStyle utilized the iterative refinement approach proposed by Alaluf et al. (2021), where the weight offsets are iteratively refined by making several passes through the HyperStyle network, improving the reconstruction quality in each pass. By default, HyperStyle performs five forward passes to invert a single image. Importantly, the changes in the generator weights do not change the latent space in a way that affects the applicability of editing directions that were found using the original generator weights. Therefore, editing directions that had been precomputed for the original generator can also be applied to the updated generator. We utilize this fact in Paper III, where we demonstrate that HyperStyle can be used in conjunction with

our proposed editing technique in order to facilitate the editing of real images.

2.5 Latent space editing

As mentioned earlier, the latent space of StyleGAN is arranged smoothly, meaning that latent codes that are close in latent space corresponds to generated images that are also similar. Further, the mapping network of StyleGAN endows the intermediate $\mathcal{W}$ space with a high level of disentanglement. As noted by Karras et al. (2019)

> "if a latent space is sufficiently disentangled, it should be possible to find direction vectors that consistently correspond to individual factors of variation." (Karras et al., 2019, p.7)

This thesis explores novel methods for finding such direction vectors that, when applied to a given latent code, results in a change in the image that can be identified as a single semantically meaningful attribute while leaving all other attributes in the image unchanged. Examples of such semantically interpretable attributes, in the context of human face portraits, include gender, age, facial expression, pose, illumination and eyeglasses. Identifying such directions is the primary prerequisite for the application of *latent-based editing*.

As a rough demarcation, the literature dealing with finding such semantically meaningful latent directions can first be divided into linear and non-linear approaches. In the remainder of this text, the term *trajectories* will be used when the path of traversal in the latent space is assumed to be non-linear, and the term *direction* will be used to denote linear direction vectors. Secondly, these special directions or trajectories can be found using either supervised or unsupervised methods.

The literature on semantic editing using StyleGAN has grown rapidly during the past few years and an exhaustive review of current methods is beyond the scope of this thesis. This section will present an overview of the work of other researchers that is most closely related to the work presented in this thesis. For a more exhaustive review of editing methods in StyleGAN, excellent survey articles exists,

see for example the work by Bermano et al. (2022), Melnik et al. (2022) and Liu et al. (2023).

Jahanian et al. (2020) proposed finding linear editing directions corresponding to edits that are easily attainable manually in the output image space, such as changes to color, zoom, 2D rotations or translations. They denote such transformations in the image space by an `edit` operation and proposed finding linear editing directions by minimizing the objective

$$\mathbf{n}^* = \arg\max_{\mathbf{n}} \mathbb{E}_{z,\alpha}[\mathcal{L}(G(\mathbf{z} + \alpha\mathbf{n}), \texttt{edit}(G(\mathbf{z}, \alpha)))], \tag{2.21}$$

where α controls the strength of the edit and $\mathcal{L}$ is an image similarity metric that they implement as either the L_2 distance or LPIPS. The authors further proposed a variation of the optimization objective to find non-linear trajectories in the latent space. Jahanian et al. (2020) noted that there are limitations to which kind of editing directions can be found using their method, and they argued that this can be explained by the distribution of the images used to train the models. For example, for a model trained on cars and for a found direction corresponding to "blueness", the authors observed that is it possible to change the color of a sportscar from red to blue by traversing the latent space in the found direction. However, when applying the same direction to a latent code corresponding to a red firetruck, the authors found that it was not possible to change the color in this case. Thus, the found latent space directions are constrained by the particular biases in the training data. To overcome this limitation, the authors proposed to augment the data set using the `edit` operation and fine-tune both the generator and editing directions on the augmented data set.

Shen et al. (2020a,b) proposed to find linear directions in a supervised fashion. This was done by first generating a collection of synthetic images and then using pretrained binary attribute classifiers to annotate the generated images according to the desired attributes. This annotation step effectively creates a labeled data set suitable for supervised learning. Shen et al. proposed to fit a Support Vector Machine (SVM) for each binary attribute, thus defining a hyperplane in the latent space corresponding to each attribute. All images corresponding to latent codes on one side of the hyperplane exhibit a particular attribute that is absent in all

images corresponding to latent codes on the other side of the hyperplane. Thus, the normal vectors to each of the supporting hyperplanes can be interpreted as semantic directions that corresponds to changes to the respective attributes. However, it was observed that moving in certain directions would also sometimes result in undesired changes in other attributes. As an example, moving in a direction corresponding to age would sometimes make eyeglasses appear in the generated image. This entanglement between age and eyeglasses can be attributed to the correlation in the training data, where older people are also more likely to be wearing eyeglasses. The authors showed that this situation could be solved by projecting the original direction onto the direction corresponding to the undesired attribute an subtracting the result from the original direction. In Paper V, we demonstrated that a similar situation occurs in the semantic latent space (Kwon et al., 2023) of diffusion models and that orthogonal projection of linear semantic directions is also an effective strategy for disentanglement for these models despite the differences in architecture.

Härkönen et al. (2020) proposed an entirely unsupervised method for finding linear semantic directions based on Principal Component Analysis (PCA) on a collection of sampled latent codes. The authors showed that this strategy yields interpretable directions in the latent space of BigGAN (Brock et al., 2019) as well as StyleGAN. As this approach is unsupervised, there is no a priori knowledge about the semantic meaning of the found directions, and therefore the effect of the found principal directions must be interpreted by manual inspection. It was observed that some of the found directions were entangled such that a single direction affected multiple semantically interpretable attributes in the output image. For example, a direction found to change the pose would also affect gender. In the context of StyleGAN, it was observed that some of this entanglement could be alleviated by only applying the found directions to certain layers of the StyleGAN synthesis network. Drawing inspiration from the work of Härkönen et al. (2020), Paper V proposes to utilize PCA to discover semantically interpretable directions in diffusion models which will be discussed in Chapter 6.

In StyleRIG, Tewari et al. (2020) proposed another supervised method aiming for full control over the head pose, facial expression, and scene illumination by using use a 3D Morphable Model (3DMM) (Blanz and Vetter, 1999). 3DMMs have

previously been widely used to model human faces. 3DMMs allow for disentangled control of a 3D face model by specifying appropriate control parameters. However, images generated by 3DMMs lack the photorealism found in StyleGAN models. In the StyleRIG framework, both the StyleGAN generator and the 3DMM remain fixed, and the goal is to learn a mapping between the parametric space of the 3DMM and the latent space of StyleGAN. The method works by learning a rigging network that takes a StyleGAN latent code $\mathbf{w} \in \mathcal{W}$ as input, along with a target semantic control parameter $\mathbf{p}$, and outputs a modified latent code $\hat{\mathbf{w}} = \text{RigNet}(\mathbf{w}, \mathbf{p})$. The modified latent code corresponds to a modified image $g(\hat{\mathbf{w}})$ that is consistent with the target control parameter while maintaining other attributes in the generated image, such as facial identity. This idea of leaning a mapping between the latent space of StyleGAN and the parameters of another model, which offers more explicit control, is similar to our proposed method in Paper III which will be presented in Chapter 4.

Wang et al. (2021) framed the application of semantic editing as a black box attack problem where the "attacker" has access to only the input and output of the model. That is, the "attacker" has access to the input latent code $\mathbf{z}\mathcal{Z}$ and the generated image $G(\mathbf{z})$, but does not have access to inspect the inner workings of the generator and it is assumed that the gradients of the generator are not available. Instead the authors assume access to one or more "victim task models" $\mathcal{M} : \mathcal{I} \rightarrow \mathcal{A}$ which map images $\mathcal{I}$ to a space of attribute scores $\mathcal{A}$. They proposed to train an proxy model $\mathcal{P} : \mathcal{Z} \rightarrow \mathcal{A}$ using sampled tuples $(\mathbf{z}, \mathcal{M}(G(\mathbf{z})))$ as a supervised training set. The gradients of the proxy models can then be calculated and a given latent code can be edited by steering though the trajectory defined by

$$\mathbf{z}^{i+1} = \mathbf{z}^{i} - \lambda \mathbf{J}_n, \tag{2.22}$$

where $\mathbf{J}_n$ is the nth row of the Jacobian of the proxy model $\mathcal{P}$ with respect to the latent code $\mathbf{z}$. The method proposed by Wang et al. is related to the method proposed in Paper III, the similarities and differences will be discussed in Section 4.3.

Shen and Zhou (2021) proposed a related approach that considers factorization of the weights of the trained generator rather than the latent codes as was proposed by Härkönen et al. (2020). In the Shen and Zhou (2021) work it is noted that the

synthesis process for an arbitrary GAN architecture can be seen as consisting of multiple layers of projection, starting from the initial latent space, and ending in the space of the final image in a series of steps. Considering the first layer which acts on the initial latent code $\mathbf{z}$, this can be written as an affine transformation

$$G_1(\mathbf{z}) \equiv \mathbf{y} = \mathbf{A}\mathbf{z} + \mathbf{b}. \tag{2.23}$$

Shen and Zhou (2021), suggests finding the direction in the latent space that induces a maximal change in the projected latent code after the step $\mathbf{y}$, which is done using the maximization objective

$$\mathbf{n}^* = \arg\max_{\mathbf{n}} \|\mathbf{A}\mathbf{n}\|_2^2 \quad \text{s.t} \quad \mathbf{n}^{\mathrm{T}}\mathbf{n} = 1. \tag{2.24}$$

In the context of StyleGAN the authors considered the transformation from latent codes to style codes. Thus, the weight matrix $\mathbf{A}$ corresponds to the concatenation of the weights of the affine transformations in the synthesis network of StyleGAN. In a follow-up work by Zhu et al. (2022) this approach was extended to allow for region-specific semantic edits by maximizing a modified objective

$$\mathbf{n}^* = \arg\max_{\mathbf{n}} \frac{\mathbf{n}^{\mathrm{T}}\mathbf{J}_f^{\mathrm{T}}\mathbf{J}_f\mathbf{n}}{\mathbf{n}^{\mathrm{T}}\mathbf{J}_b^{\mathrm{T}}\mathbf{J}_b\mathbf{n}} \quad \text{s.t} \quad \mathbf{n}^{\mathrm{T}}\mathbf{n} = 1, \tag{2.25}$$

where $\mathbf{J}_f$ and $\mathbf{J}_b$ are the Jacobians of G with respect to the latent code, for the foreground (region selected by the user) and background (the complement of the selected region), respectively. This approach enables location-specific edits like opening or closing the eyes, modifying the mouth, or changing the density of the eyebrows. However, Zhu et al. (2022) pointed out that the method was not able to control individual elements of symmetric pairs within the image; for example, the method is unable to close only one eye while keeping the other open. Further, this approach is not able to change global attributes that are not confined to a well-defined region in the image, such as changes to pose, age and gender.

Rather than editing in the $\mathcal{Z}$, $\mathcal{W}$ or $\mathcal{W}+$ spaces, Wu et al. (2021) proposed using a different latent space for semantic editing in StyleGAN. The proposed *style space*, denoted as $\mathcal{S}$ space, is the space spanned by the outputs of the learned

affine transformations in the synthesis network. The authors presented a method to automatically discover directions in $\mathcal{S}$ space that control a single attribute in the generated image. Specifically, they propose calculating the gradient map of the generated images with respect to each dimension of the style codes $\mathbf{s} \in \mathcal{S}$. These gradient maps were then compared to semantic maps obtained from a pre-trained segmentation network. The aim is to identify style space channels where the corresponding gradient maps have a high overlap with the segmentation maps corresponding to a single specific semantic label. Wu et al. (2021) showed that this method successfully identifies style space channels that control highly localized attributes in the generated images such as the color and style of the hair, shape of the ears, eyes, and eyebrows as well as gaze direction.

Chapter 3

A multilinear model for faces

This chapter serves to introduce Paper I and Paper II. The central idea in these publications is to use a multilinear model to factorize the latent space of StyleGAN into semantically meaningful subspaces, thus allowing for more explicit control over the generated images.

Concretely, we used a labeled facial expression data set of real face images which we initially project into the latent space of StyleGAN. We then used a tensor decomposition method to factorize the projected data according to the provided labels. In this way we discover subspaces in the StyleGAN latent space which corresponds to each of the six prototypical expressions, happiness, surprise, anger, sadness, fear, and disgust as well as a subspace that controls the yaw rotation of the generated image.

This chapter begins in Section 3.1 with an overview of basic definitions and operations from multilinear algebra that are used in the papers. Next, Section 3.2 will review related work where multilinear tensor models have been used to model human faces and expressions in contexts other than DGMs. Finally, Section 3.3 will summarize the main findings of the papers.

3.1 Multilinear algebra

Tensors are multidimensional arrays, a generalization of vectors and matrices. The *order* a tensor specifies the number of indices required to uniquely specify an element of the tensor. From this definition, we can associate first- and second-order tensors with vectors and matrices respectively. Tensors with order three or higher are called higher-order tensors. In the remainder of this text, the word tensor is reserved for these higher-order tensors. While linear algebra involves the study of vectors and matrices, multilinear algebra studies tensors and operations on tensors.

Fibers are the higher-order analog to the notion of matrix rows and columns. We can extract the mode-n fiber of a tensor by fixing all but the nth index. For example, consider the third-order tensor $T \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ with components T_{ijk}, then the mode-2 fiber of T is the vector obtained by selecting a particular i and k while leaving j as a free index. Kolda and Bader (2009) denoted this with the "matlab-esque" notation $\mathbf{t}_{i:k} \in \mathbb{R}^{I_2}$. Higher-order analogs to fibers can also be defined, for example, *slices* are two-dimensional sections of a tensor defined by fixing all but two indices. In the notation of Kolda and Bader, the third-order tensor T would have $\mathbf{T}_{i::} \in \mathbb{R}^{I_2 \times I_3}$, $\mathbf{T}_{:j:} \in \mathbb{R}^{I_1 \times I_3}$ and $\mathbf{T}_{::k} \in \mathbb{R}^{I_1 \times I_2}$ as the possible slices.

The mode-n unfolding, or matricization of a tensor is essentially a reshaping operation that transforms a tensor $T \in \mathbb{R}^{I_1 \times \cdots \times I_n \times \cdots \times I_N}$ into a matrix $\mathbf{T}_{(n)} \in \mathbb{R}^{I_n \times I_1 \cdots I_{n-1} I_{n+1} \cdots I_N}$ by arranging the mode-n fibers as the rows and all other rearranged as the columns of the matrix. The mode unfolding operation is intuitively easy to understand since one just moves the relevant mode index of the tensor to become the first index of the resultant matrix, and concatenates all other modes of the tensor into the second mode of the matrix. A graphical illustration of the unfolding operation is provided in Figure 3.1. However the exact definition of the necessary index permutation is a bit clunky. In fact, there are several ways that the unfolding operation can be defined (Kossaifi, 2017). Here we follow Kolda and Bader (2009) and define the index permutation on the unfolding operation in such

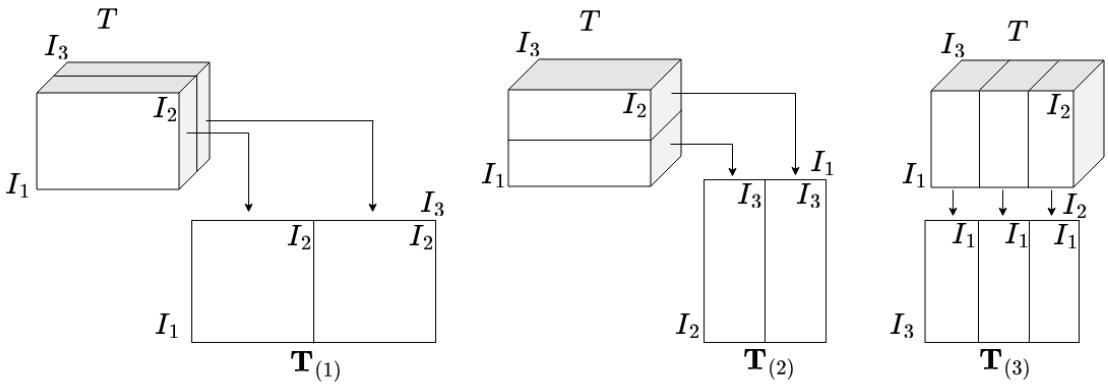


Figure 3.1: **Graphical illustration of the mode-n unfolding.** The mode-n unfolding is an operation that transforms a tensor into a matrix. An order 3 tensor $T \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, has three possible unfoldings, $\mathbf{T}_{(1)} \in \mathbb{R}^{I_1 \times I_2 I_3}$, $\mathbf{T}_{(2)} \in \mathbb{R}^{I_2 \times I_3 I_1}$ and $\mathbf{T}_{(3)} \in \mathbb{R}^{I_3 \times I_1 I_2}$. The figure was made with inspiration from the work of Vasilescu and Terzopoulos (2002).

a way that the tensor element $T_{i_1 \cdots i_n \cdots, i_N}$ maps to the matrix element $(\mathbf{T}_{(n)})_{i_n j}$ with

$$j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_k - 1) J_k \qquad \text{with} \qquad J_k = \prod_{\substack{m=1 \\ m \neq n}}^{k-1} I_m. \tag{3.1}$$

The mode-n product is an operation that multiplies a tensor by a matrix. For a general order N tensor $T \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n \times \cdots \times I_N}$ the mode-n product of T with a matrix $\mathbf{A} \in \mathbb{R}^{J_n \times I_n}$ is defined as

$$(T \times_n \mathbf{A})_{i_1, i_2, \cdots, i_{n-1}, j_n, i_{n+1}, \cdots, i_N} = \sum_{i_n=1}^{I_n} T_{i_1, i_2, \cdots, i_{n-1}, i_n, i_{n+1}, \cdots, i_N} a_{j_n, i_n} \tag{3.2}$$

with $T \times_n \mathbf{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times J_n \times \cdots \times I_N}$. The mode-$n$ product is commutative when the matrices are applied along distinct modes, *i.e.*,

$$(T \times_n \mathbf{A}) \times_m \mathbf{B} = (T \times_m \mathbf{B}) \times_n \mathbf{A} \tag{3.3}$$

for $\mathbf{A} \in \mathbb{R}^{J_n \times I_n}$ and $\mathbf{B} \in \mathbb{R}^{J_m \times I_m}$ when $n \neq m$.

The Higher-Order Singular Value Decomposition (HOSVD) or Tucker decomposi-

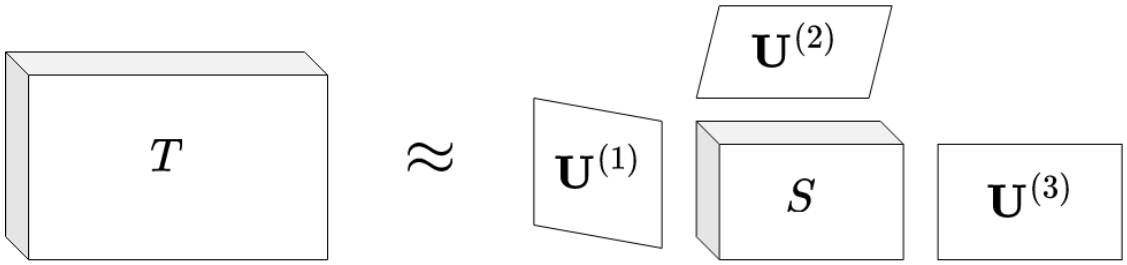


Figure 3.2: **Graphical illustration of the HOSVD/Tucker decomposition.** The HOSVD is a generalization of the matrix SVD which composes an arbitrary tensor T into a core tensor S and a set of factor matrices $\mathbf{U}^{(i)}$.

tion (Tucker, 1966) can be seen as a generalization of the well-known matrix SVD to higher-order tensors. The HOSVD factorizes an arbitrary tensor T into a core tensor S and a set of orthogonal factor matrices $\mathbf{U}^{(n)}$, one for each mode of the tensor T. Thus the HOSVD of an order-N tensor consists of a single core tensor and N factor matrices. As an example consider the HOSVD of the order-3 tensor $T \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. The HOSVD of T can be written as

$$T = S \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)}, \tag{3.4}$$

where $S \in \mathbb{R}^{J_1, \times J_2 \times J_3}$ is the core matrix and $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times J_n}$ are the corresponding factor matrices. An intuitive graphical illustration of the HOSVD operation is shown in Figure 3.2. The algorithm to calculate the HOSVD is relatively simple and can be defined by calculating the ordinary matrix SVD for each of the mode-n unfoldings of the tensor T. The factor matrices are then the left hand singular vectors of the mode-n unfoldings and the core tensor can be calculated by applying the transpose of each factor matrix to the tensor via the mode-n product. The full algorithm is provided in Algorithm 1.

An order-N tensor is *rank one* if it can be written as an outer product of N vectors

$$T = \mathbf{t}^{(1)} \otimes \mathbf{t}^{(2)} \otimes \cdots \otimes \mathbf{t}^{(N)}, \tag{3.5}$$

where $\otimes$ is the vector outer product which can be defined component-wise as

$$(\mathbf{t}^{(1)} \otimes \mathbf{t}^{(2)} \otimes \cdots \otimes \mathbf{t}^{(N)})_{i_1 i_2 \cdots i_N} = t^{(1)}_{i_1} t^{(2)}_{i_2} \cdots t^{(N)}_{i_N} \tag{3.6}$$

Algorithm 1 Higher-order Singular Value Decomposition (HOSVD)

Input: $T \in \mathbb{R}^{I_1 \times \cdots \times I_n \times \cdots \times I_N}$
Output: $(S \in \mathbb{R}^{J_1 \times \cdots \times J_n \times \cdots \times J_N}, \mathbf{U}^{(n)} \in \mathbb{R}^{J_n \times I_n})$ - Core tensor and factor matrices.
  for $n = 1, \cdots, N$ **do**
    $\mathbf{T}_{(n)} \leftarrow \text{Unfold}_n(T)$ ▷ Mode-n unfolding of T
    $\mathbf{U}^{(n)}, \mathbf{\Sigma}^{(n)}, \mathbf{V}^{(n)} \leftarrow \text{SVD}(\mathbf{T}_{(n)})$ ▷ Matrix SVD of the mode-n unfolding
  end for
  $S \leftarrow T \times_1 \mathbf{U}^{(1)\text{T}} \times_2 \mathbf{U}^{(2)\text{T}} \cdots \times_N \mathbf{U}^{(N)\text{T}}$

Any tensor can be defined as a sum of rank one tensors as

$$T = \sum_{r=1}^{R} \mathbf{t}_r^{(1)} \otimes \mathbf{t}_r^{(1)} \otimes \cdots \otimes \mathbf{t}_r^{(N)}. \tag{3.7}$$

The rank of a tensor is defined as the minimal number of rank one tensors needed to reconstruct it. If R is minimal, then a decomposition of the form of Eq. (3.7) is also called the Canonical Polyadic Decomposition (Kolda and Bader, 2009).

The HOSVD can be used for dimensionality reduction of tensors in a way similar to how PCA is used to reduce the dimensionality of matrices. By truncating the factor matrices in the HOSVD, we can reduce the dimensionality of each of the modes. By selecting only the first $\tilde{I}_n$, with $\tilde{I}_n \leq I_n$, dominant mode singular vectors contained in each of the factor matrices $\mathbf{U}^{(n)}$ we can calculate truncated core tensor $\tilde{S}$ and obtain an approximation of T denoted as $\hat{T}$ as

$$T \approx \hat{T} = \tilde{S} \times_1 \widetilde{\mathbf{U}}^{(1)} \times_2 \widetilde{\mathbf{U}}^{(2)} \times_3 \widetilde{\mathbf{U}}^{(3)}, \tag{3.8}$$

with $T \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, $S \in \mathbb{R}^{\tilde{I}_1 \times \tilde{I}_2 \times \tilde{I}_3}$ and $\widetilde{\mathbf{U}^{(1)}} \in \mathbb{R}^{I_1 \times \tilde{I}_1}$. The error of the approximation is bounded by the sum of squared singular values associated with the discarded singular vectors (Vasilescu and Terzopoulos, 2003)

$$||T - \hat{T}||_F^2 \leq \sum_{i_1} \sigma_{i_1}^2 + \sum_{i_2} \sigma_{i_2}^2 + \cdots \sum_{i_n} \sigma_{i_n}^2. \tag{3.9}$$

In Paper II, we use this technique to define a linear expression intensity subspace by truncating the mode of the data tensor corresponding to expression intensity up to the dominant mode singular vector.

3.2 Related Work

The HOSVD has been used as a central tool for modeling faces and expressions in multiple prior works. Vasilescu and Terzopoulos (2002) used a model based in the HOSVD to analyse face images from the Weizmann face database (Moses et al., 1996). The database contains 512×352 grayscale images of 28 male subjects from 5 different pose angles, 3 different illumination conditions, and 3 different expressions. Vasilescu and Terzopoulos performed a multilinear analysis on this data set utilising the HOSVD to decompose the data tensor $T \in \mathbb{R}^{28\times5\times3\times3\times(512\cdot352)}$ according to the different modes of variation as

$$T = S \times_1 \mathbf{U}_{\text{people}} \times_2 \mathbf{U}_{\text{views}} \times_3 \mathbf{U}_{\text{illums}} \times_4 \mathbf{U}_{\text{expres}} \times_5 \mathbf{U}_{\text{pixel}}. \tag{3.10}$$

The authors note that the multilinear treatment has significant advantages over and subsumes the conventional PCA treatment and note that the images contained as the columns of $\mathbf{U}_{\text{pixel}}$ are identical to the more conventional eigenfaces (Sirovich and Kirby, 1987). This point is easy to see since $\mathbf{U}_{\text{pixel}}$ contains the left singular vectors of the particular mode unfolding of T which exactly matches the design matrix containing the pixels in each row and each data point along the columns. The left-hand singular vectors of the design matrix are identical to the principal components derived from PCA

Graßhof et al. (2017) used a HOSVD-based tensor model to analyse 3D facial feature points of the Binghamton University 3D Facial Expression (BU-3DFE) (Yin et al., 2006) database. They used 83 3D facial feature points stemming from scans of 100 subjects performing 25 different expressions. They performed a HOSVD on the data in the space of 3D point shapes and proposed to model data points by selecting an appropriate linear combination of the mode singular vectors as

$$\hat{\mathbf{w}} = \mathbf{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{p}_2^{\mathrm{T}}\mathbf{U}^{(2)} \times_3 \mathbf{p}_3^{\mathrm{T}}\mathbf{U}^{(3)}, \tag{3.11}$$

where $\mathbf{p}_2$ and $\mathbf{p}_3$ are parameter vectors corresponding to the person and expression respectively and $\mathbf{U}^{(2)}$ and $\mathbf{U}^{(3)}$ contains the basis vectors for the person and expression subspace respectively. By analyzing the expression subspace determined

by $\mathbf{U}^{(3)}$, the authors found that this expression feature space had a "star-shaped" structure and that expression trajectories intersect in a unique point which they regarded as the origin point of the expression subspace. This point is different from the neutral expression which is present in the data set, and the authors coined this special point the "point of apathy" due to the emotionless apathetic facial expression generated by selecting this point as the coefficient $\mathbf{p}_3^{\mathrm{T}}$ in Eq. (3.11).

In order to estimate the model the parameters $\mathbf{p}_2$ and $\mathbf{p}_3$ of a new previously unseen shape, Graßhof et al. proposed the minimization problem

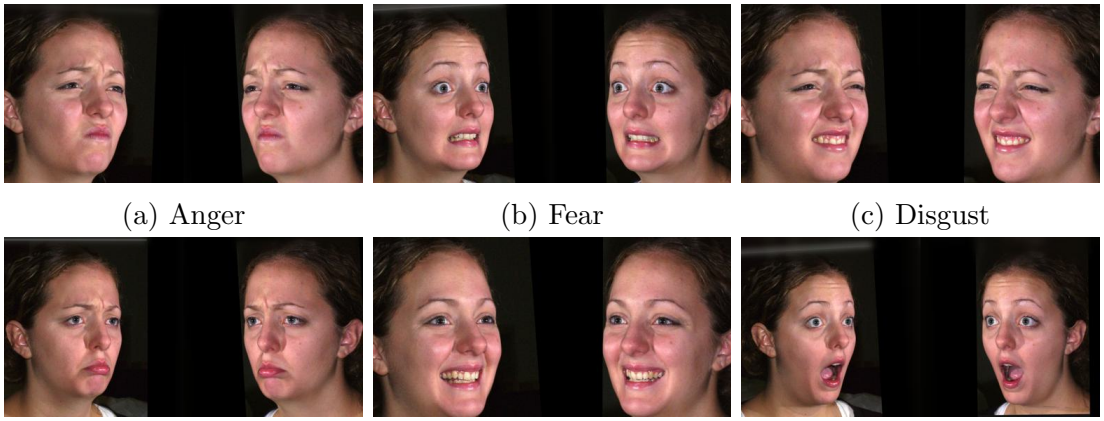
$$\begin{aligned}\min_{\mathbf{p}_2,\mathbf{p}_3} \|\hat{\mathbf{w}} - \mathbf{w}\| + \lambda_1 \|\mathbf{p}_2\|_2^2 + \lambda_2 \left\|\mathbf{p}_2^{\mathrm{T}}\mathbf{1} - 1\right\|_2^2 \\ + \lambda_3 \|\mathbf{p}_3\|_2^2 + \lambda_4 \left\|\mathbf{p}_3^{\mathrm{T}}\mathbf{1} - 1\right\|_2^2,\end{aligned} \tag{3.12}$$

which can be solved using an alternating least squares method which is also explained in detail in Paper I. Further, Graßhof et al. (2017) experimented with using the tensor model for expression classification by estimating $\mathbf{p}_3$, the parameter corresponding to facial expressions, using Eq.(3.12).

3.3 Tensor-based expression editing

The main aim of Paper I and Paper II is to gain greater control over the images synthesized by StyleGAN enabling the synthesis of face images with a specific pose or facial expression. The central idea in the papers is to use a HOSVD-based tensor model to factorize the latent space of StyleGAN into several, semantically meaningful, subspaces. These subspaces control the identity, expression, and rotation of the generated faces respectively. This is the first time a HOSVD-based tensor model has been used in the context of state-of-the-art DGMs like StyleGAN.

To construct a HOSVD-based tensor model for the StyleGAN latent space we use a real facial expression data set as supervision. As in Graßhof et al. (2017) we use the BU-3DFE database, but use the raw images rather than the 3D facial feature points. The data consists of images of 100 different persons, 56 females, and 44 males, with good coverage of different ages (18-70 years) and ethnicity. Each subject was asked to perform a selection of different facial expressions in front of a



(a) Anger (b) Fear (c) Disgust

(d) Sadness (e) Happiness (f) Surprise

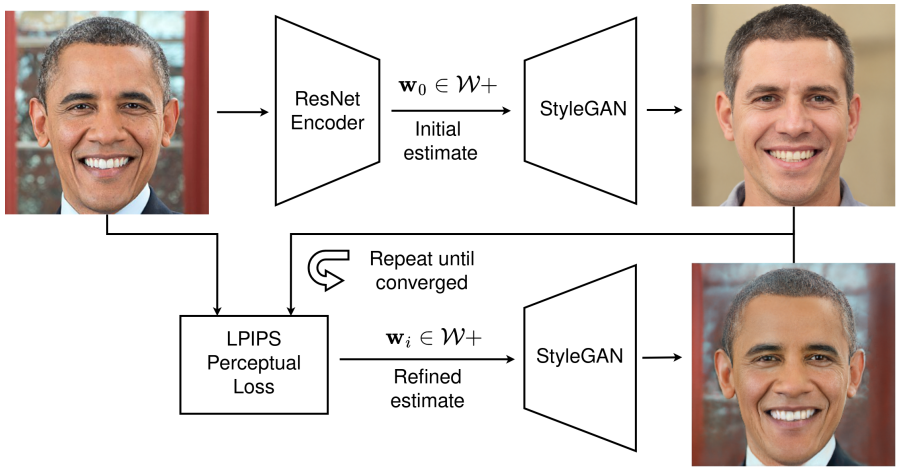
Figure 3.3: **Samples from BU-3DFE database.** The BU-3DFE database consists of real images of 100 different people each performing the six prototypical expressions at various levels of intensity. The figure shows the images corresponding to the most intense expressions for the first person in the data set.

3D face scanner. Concretely, the subjects were asked to perform a single natural expression as well as each of the six prototypical expressions (happiness, disgust, fear, anger, surprise, and sadness). Each of the prototypical expressions was performed at four different levels of intensity. The participants of the database were undergraduates, graduates, and faculty from Binghamton University's departments of psychology, arts, and engineering. Each of the performed facial expressions was captured at two view views (about +45° and -45° from a frontal angle). With 100 unique identities performing 25 expressions from two views, the data set contains 5000 unique images. Figure 3.3 shows the raw (aligned) data for the most intense expressions for the first person in the data set.

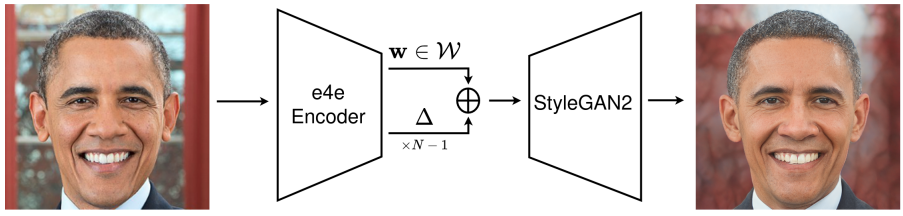
The first step for constructing a tensor model from the BU-3DFE data set is to project each image into the latent space of StyleGAN to obtain a latent representation for each of the data points. In Paper I we used a hybrid two-step inversion technique where a ResNet encoder was first trained[1] to provide a good initial estimate of the latent code $\mathbf{w} \in \mathcal{W}+$ which is subsequently refined using LPIPS perceptual loss. An illustration of this approach is shown in Figure 3.4a.

Although this hybrid approach gave good initial reconstructions of the images

[1]Training code is provided by Baylies. (2019)



(a) Illustration of projection method in Paper I



(b) Illustration of projection in Paper II

Figure 3.4: **Illustration of GAN inversion methods used in the papers.** (a) In Paper I we used a two-step inversion technique where a ResNet-based encoder is first used to obtain a rough estimate of a latent code which is subsequently refined using LPIPS as a perceptual loss. (a) In Paper II we used the e4e encoder proposed by Tov et al. (2021) which ensures that the found latent codes reside in an editable resion of the latent space.

from BU-3DFE and the method can define latent directions corresponding to yaw rotation, editing expressions was more challenging. We attribute this to the lower editability of the latent codes according to the distortion-editability trade-off which was introduced in Section 2.5. In Paper II, we used the e4e encoder (Tov et al., 2021) to obtain a latent representation of the BU-3DFE data set, which had a good trade-off between distortion and editability. An illustration of the GAN inversion using the e4e encoder is shown in Figure 3.4b.

We project the images from BU-3DFE into the $\mathcal{W}+$ space of StyleGAN, thus representing each image by a latent code of dimension (18×512). We flatten each of the latent codes in $\mathcal{W}+$ into vectors $\text{vec}(\mathbf{w}) \in \mathbb{R}^{9216}$. In Paper I, we then arrange

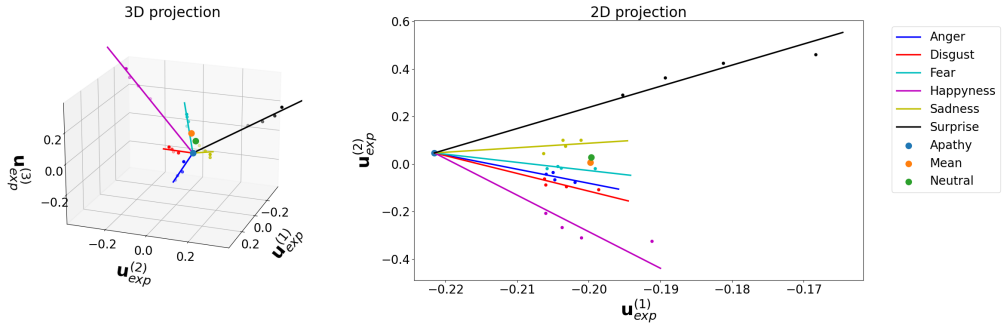


Figure 3.5: **Visualizing the expression subspace.** We see that each of the six prototypical expressions follows approximately linear trajectories in the latent space. Extrapolating the trajectories indicates that they have a common origin in the "point of apathy", which is different from both the average and neutral expression.

the data in a fourth-order data tensor T such that each mode of T corresponds to a single factor of variation. In this case, the first mode corresponds to the mean-centered latent codes, the second mode corresponds to each of the 100 person identities, the third mode contains the 25 expressions and finally, the fourth mode contains the 2 rotations.

The HOSVD of the mean-centered data tensor $T - \bar{T} \in \mathbb{R}^{9216\times100\times25\times2}$, where $\bar{T}$ is the tensor with the mean latent code $\bar{\mathbf{w}}$[2] repeated in all entries, can be written as

$$T - \bar{T} = S \times_1 \mathbf{U}_{\text{lat}} \times_2 \mathbf{U}_{\text{per}} \times_3 \mathbf{U}_{\text{exp}} \times_4 \mathbf{U}_{\text{rot}}. \tag{3.13}$$

In Paper I, we investigated the structure of expression subspace. The columns of $\mathbf{U}_{\text{exp}}$ define a basis for the 25-dimensional expression subspace. Figure 3.5 shows 2D and 3D projections of how the data points from the BU-3DFE data set are distributed in the expression subspace. We observe the same "star-shaped" structure that was reported by Graßhof et al. (2017) even though we here use projected latent codes rather than 3D point features.

In Paper II, we consider ordering the expression intensities into a dedicated mode of the data tensor by discarding the neural expression and reshaping the data tensor

[2]In this context, the mean latent code $\bar{\mathbf{w}}$ refers to the mean of the projected latents from the BU-3DFE data set, rather than the mean of the distribution of the pretrained generator.

from an order 4 to an order 5 tensor $T \in \mathbb{R}^{9216\times100\times6\times4\times2}$ woth HOSVD

$$T - \bar{T} = S \times_1 \mathbf{U}_{\text{lat}} \times_2 \mathbf{U}_{\text{per}} \times_3 \mathbf{U}_{\text{exp}} \times_4 \mathbf{U}_{\text{int}} \times_5 \mathbf{U}_{\text{rot}}. \tag{3.14}$$

We can introduce relevant subspace parameters $\mathbf{q}_{\text{per}} \in \mathbb{R}^{100}$, $\mathbf{q}_{\text{exp}} \in \mathbb{R}^{6}$, $\mathbf{q}_{\text{int}} \in \mathbb{R}^{4}$, $\mathbf{q}_{\text{rot}} \in \mathbb{R}^{2}$, controlling identity, expression, expression intensity, and rotation respectively. These subspace parameters pick out the appropriate linear combination from each of the factor matrices to reconstruct individual data points as

$$\mathbf{w} = \bar{\mathbf{w}} + S \times_1 \mathbf{U}_{\text{lat}} \times_2 \mathbf{q}_{\text{per}}^{\text{T}}\mathbf{U}_{\text{per}} \times_3 \mathbf{q}_{\text{exp}}^{\text{T}}\mathbf{U}_{\text{exp}} \times_4 \mathbf{q}_{\text{int}}^{\text{T}}\mathbf{U}_{\text{int}} \times_5 \mathbf{q}_{\text{rot}}^{\text{T}}\mathbf{U}_{\text{rot}}. \tag{3.15}$$

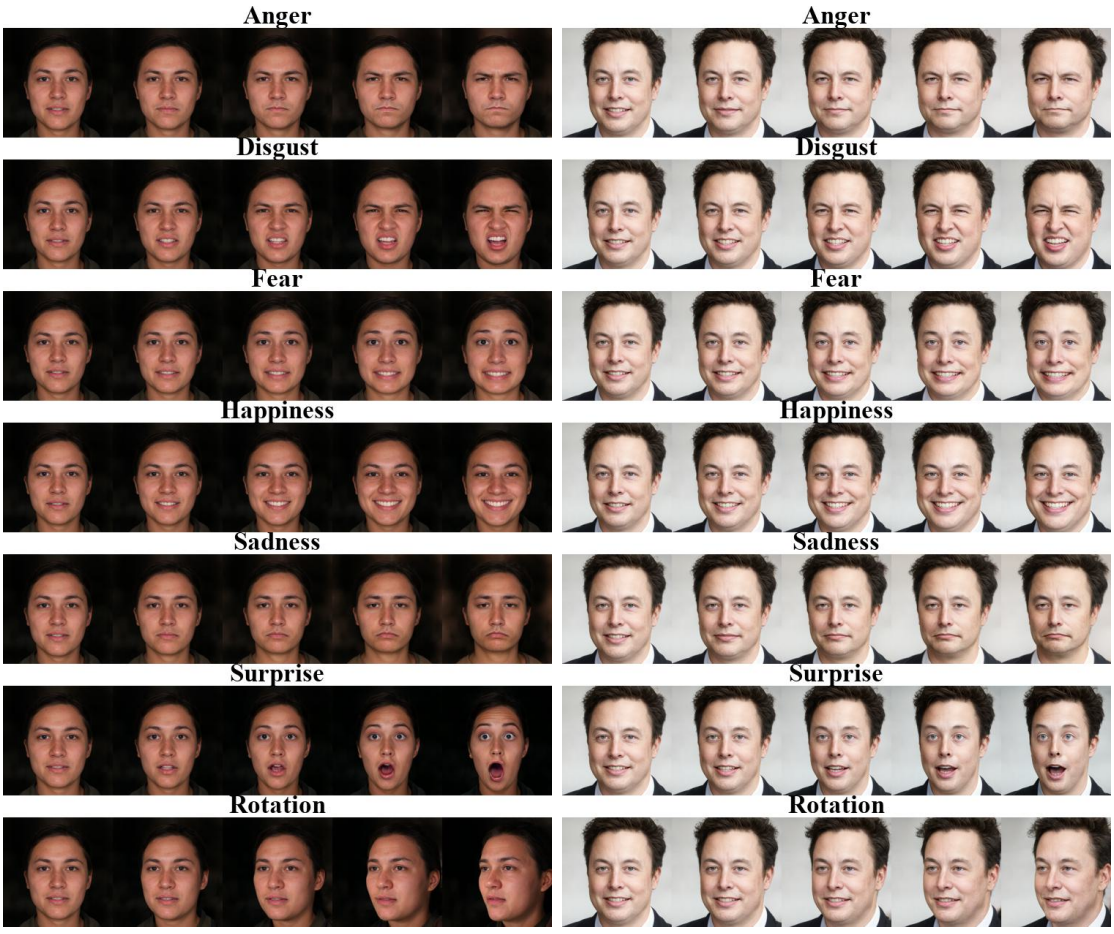
In Paper I it was assumed that the subspace parameters should first be estimated to perform semantic editing on latent codes which were not a part of the training data. On the contrary, in Paper II we propose to use the tensor model formulation to extract global semantic directions that can be applied to any latent code without the need for estimating the parameters of the tensor model beforehand.

To find such global editing directions corresponding to the six prototypical expressions, we truncate the expression intensity subspace to the dominant mode singular vector by selecting only the first column of $\mathbf{U}_{\text{int}}$ which is denoted at $\mathbf{u}_{\text{int}}$. We then select the average parameters associated with identity and rotation, $\bar{\mathbf{q}}_{\text{per}}$ and $\bar{\mathbf{q}}_{\text{rot}}$ respectively, and global editing directions $\mathbf{n}_{\text{expr}}$ can then be defined as

$$\mathbf{n}_{\text{expr}} = S \times_1 \mathbf{U}_{\text{lat}} \times_2 \bar{\mathbf{q}}_{\text{per}}^{\text{T}}\mathbf{U}_{\text{per}} \times_3 \mathbf{q}_{\text{exp}}^{\text{T}}\mathbf{U}_{\text{exp}} \times_4 \mathbf{q}_{\text{int}}^{\text{T}}\mathbf{u}_{\text{int}} \times_5 \bar{\mathbf{q}}_{\text{rot}}^{\text{T}}\mathbf{U}_{\text{rot}}. \tag{3.16}$$

Editing can then be performed by linearly perturbing the latent codes in the direction of the expression directions as $\mathbf{w}_{\text{edit}} = \mathbf{w}_0 + \gamma\mathbf{n}_{\text{expr}}$ where γ controls the intensity of the edit.

Figure 3.6 shows the result of interpolating along semantic directions found using Eq. (3.16). In Figure 3.6a the directions are applied to the average face of the BU-3DFE data set and in Figure 3.6b they are applied to a real image of Elon Musk that has been projected into the latent space of StyleGAN2 using the e4e encoder (Tov et al., 2021).



(a) BU-3DFE "mean face". (b) Projected image of Elon Musk.

Figure 3.6: **The effect of global editing directions**. (a) Shows the effect of interpolation along the found linear editing directions for the mean face of the BU-3DFE data set. (b) Shows the effect of applying the same semantic directions to an image of Elon Musk that has been projected into the latent space using the e4e encoder (Tov et al., 2021).

Chapter 4

Controllable synthesis using NRSfM

This chapter will introduce Paper III which proposes a simple and efficient method to control the 3D geometry of face images synthesized by StyleGAN. Since the StyleGAN generator has been exclusively trained on 2D images, it does not have any explicit knowledge about the 3D structure of the objects that it can generate.

In Paper III, we propose to use Non-Rigid Structure-from-Motion (NRSfM) as a simple framework for endowing StyleGAN with explicit 3D control of the geometry of the synthesized faces. This allows us to both predict the 3D structure directly from the latent codes and modify the latent codes such that the generated 2D images are consistent with a specified 3D geometry. Paper III provides the first method for combining NRSfM, a classical problem in computer vision, with a modern DGM architecture like StyleGAN.

The chapter begins in Section 4.1 with an introduction to the NRSfM problem. Next, Section 4.2 introduces the method proposed in Paper III that connects a sparse face model based on NRSfM with the latent space of StyleGAN. Finally, Section 4.3 discusses Paper III in relation to other research related to semantic editing and methods for gaining explicit 3D control of face images generated with StyleGAN.

4.1 Non-rigid structure from motion

The Structure-from-Motion (SfM) problem is a classical problem in computer vision and dates back to the early nineties with the seminal work of Tomasi and Kanade (1992). The aim of SfM is to reconstruct the 3D geometry of a scene consisting of non-deformable objects using a sequence of corresponding 2D points as the input. In the traditional formulation of the problem we have N corresponding points in each of F frames. The Tomasi-Kanade factorization method seeks to factorize a set of observations, which are collected into a measurement matrix $\mathbf{W} \in \mathbb{R}^{2F\times N}$, in terms of a motion matrix $\mathbf{M} \in \mathbb{R}^{2F\times 3}$ and a structure matrix $\mathbf{S} \in \mathbb{R}^{3\times N}$ as

$$\mathbf{W} = \mathbf{M}\mathbf{S}, \tag{4.1}$$

where the structure matrix $\mathbf{S}$ contains the 3D world coordinates for the object of the scene and the motion matrix $\mathbf{M} = \begin{bmatrix}\mathbf{M}_1 & \mathbf{M}_2 & \cdots & \mathbf{M}_F\end{bmatrix}^{\mathrm{T}}$ contains the 2×3 projection matrices $\mathbf{M}_i$ for each of the given F frames.

The factorization in Eq. (4.1) can be performed by arguing the that the measurement matrix $\mathbf{W}$ must be rank 3. Recall that the rank of the product of two matrices is constrained by the rank of the factors. In other words, for two arbitrary matrices $\mathbf{A} \in \mathbb{R}^{n\times m}, \mathbf{A} \in \mathbb{R}^{m\times l}$ we have

$$\mathrm{rank}(\mathbf{AB}) \leq \min(\mathrm{rank}(\mathbf{A}), \mathrm{rank}(\mathbf{B})). \tag{4.2}$$

This implies that the measurement matrix $\mathbf{W}$ must have $\mathrm{rank}(\mathbf{W}) = 3$ if $N, F \geq 3$. For this reason the Singular Value Decomposition (SVD) (Golub and Reinsch, 1970) of $\mathbf{W}$, written as $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$ has at most 3 non-zero singular values. Now assume that the singular values are ordered in decreasing order and let $\mathbf{U}_0$ denote the first 3 columns of $\mathbf{U}$, $\mathbf{\Sigma}_0$ the first 3-by-3 submatrix of $\mathbf{\Sigma}$ and $\mathbf{V}_0^{\mathrm{T}}$ he first 3 rows $\mathbf{V}^{\mathrm{T}}$ Then a valid decomposition of Eq. (4.1) is

$$\mathbf{W} = \underbrace{\mathbf{U}_0\mathbf{\Sigma}_0^{1/2}}_{\mathbf{M}}\underbrace{\mathbf{\Sigma}_0^{1/2}\mathbf{V}_0^{\mathrm{T}}}_{\mathbf{S}}. \tag{4.3}$$

However, this decomposition is not unique since for any invertible matrix $\mathbf{Q}$ the

decomposition $\mathbf{M}' = \mathbf{MQ}$ and $\mathbf{S}' = \mathbf{Q}^{-1}\mathbf{S}$ is also valid since it leaves $\mathbf{W}$ unchanged

$$\mathbf{M}'\mathbf{S}' = (\mathbf{MQ})(\mathbf{Q}^{-1}\mathbf{S}) = \mathbf{M}(\mathbf{QQ}^{-1})\mathbf{S} = \mathbf{MS} = \mathbf{W}. \tag{4.4}$$

In practice, this ambiguity can be solved by imposing additional constraints on the factorization in Eq. (4.1). This can be done by demanding that the rows of $\mathbf{M}_i$ have unit norm and are orthogonal to each other, *i.e.* by demanding that $\mathbf{M}_i\mathbf{M}_i^{\mathrm{T}} = \mathbf{I}_2$, in which case the factorization is unique up to an arbitrary rotation (Trucco and Verri, 1998, p.207).

In SfM it is assumed that the reconstructed object is rigid and non-deformable. This is a strong limitation as many real-world objects are deformable and might change. This is particularly true for human faces where, for example, different facial expressions are highly non-rigid deformations to the underlying 3D shape. If we wish to model complex non-rigid deformations such as human facial expressions we will need to extend SfM to allow for non-rigid deformations.

The NRSfM problem seeks to find the 3D reconstruction of scenes where the object is allowed to undergo non-rigid deformations. The first method for solving the NRSfM problem is often attributed to Bregler et al. (2000). The fundamental assumption of NRSfM is that any 3D shape can be expressed as a rigid basis-shape $\mathbf{B}_0$ and a linear combination of a finite set of K non-rigid basis shapes $\{\mathbf{B}_i\}_{i=1}^{K}$. The rigid basis-shape describes the average reconstructed 3D shape and each of the non-rigid basis shapes model the variation from this average shape. Thus, the resultant 3D shape under an arbitrary 3D deformation can be written as

$$\mathbf{S} = \mathbf{B}_0 + \sum_{i=1}^{K} \alpha_i \mathbf{B}_i. \tag{4.5}$$

Using this assumption we see that the measurement matrix $\mathbf{W}$ can be written as

$$\mathbf{W} = \underbrace{\begin{bmatrix} \mathbf{M}_1 & \alpha_{11}\mathbf{M}_1 & \alpha_{12}\mathbf{M}_1 & \cdots & \alpha_{1K}\mathbf{M}_1 \\ \mathbf{M}_2 & \alpha_{21}\mathbf{M}_2 & \alpha_{22}\mathbf{M}_2 & \cdots & \alpha_{2K}\mathbf{M}_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{M}_F & \alpha_{F1}\mathbf{M}_F & \alpha_{F2}\mathbf{M}_F & \cdots & \alpha_{FK}\mathbf{M}_F \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_1 \\ \mathbf{B}_2 \\ \vdots \\ \mathbf{B}_K \end{bmatrix}}_{\mathbf{B}}, \tag{4.6}$$

where the $2F \times N$ measurement matrix $\mathbf{W}$ is be decomposed as a $3(K+1) \times N$ structure matrix $\mathbf{B} = \begin{bmatrix} B_0 & B_1 & B_2 & \cdots & B_K \end{bmatrix}^{\mathrm{T}}$ and a $2F \times 3(K+1)$ motion matrix $\mathbf{M}$. Here the motion matrix can be recovered by first computing the SVD of the measurement matrix as $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$ and then defining $\mathbf{M}$ as the first $3(K+1)$ columns of $\mathbf{U}$, each multiplied by their respective singular values as $\mathbf{M} = \begin{bmatrix} \sigma_1\mathbf{u}_1 & \sigma_2\mathbf{u}_2 & \cdots & \sigma_{3(K+1)}\mathbf{u}_{3(K+1)} \end{bmatrix}$ (Hartley and Zisserman, 2004, p.444).

As in the in rigid case, the non-rigid factorization in Eq. (4.6) is not unique since the insertion of an invertible $3(K+1) \times 3(K+1)$ matrix $\mathbf{Q}$ leaves the measurement matrix $\mathbf{W}$ unchanged since $\mathbf{W} = \mathbf{M}\mathbf{B} = \mathbf{M}\mathbf{Q}\mathbf{Q}^{-1}\mathbf{B}$. However, in the non-rigid case it is an additional requirement that the matrix $\mathbf{Q}$ is chosen such that $\mathbf{M}\mathbf{Q}$ has the block structure of Eq. (4.6).

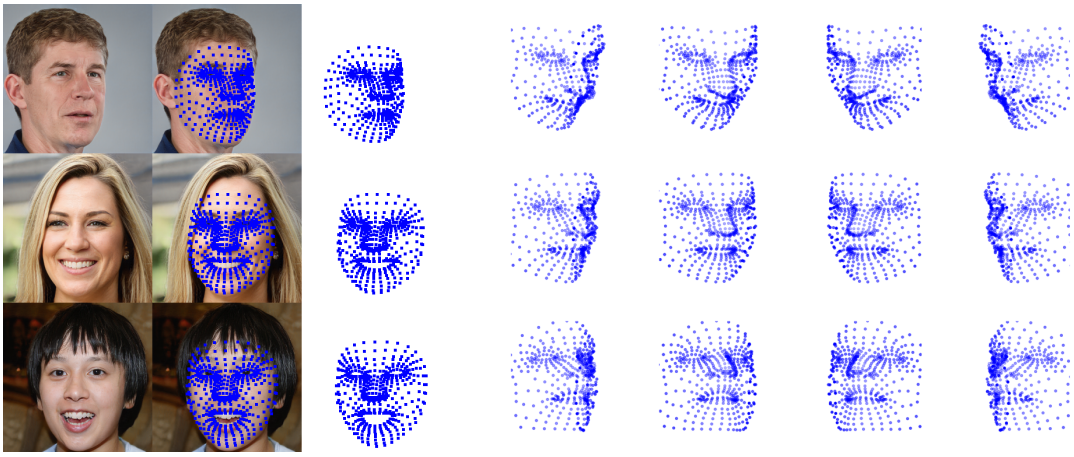
In Paper III we used the NRSfM method proposed by Brandt and Ackermann (2019) which proposed to assume that the non-rigid basis shapes are rank-one *i.e.* can be written as $\mathbf{B}_i = \mathbf{d}_i\mathbf{b}_i^{\mathrm{T}}$ for some vectors $\mathbf{d}_i \in \mathbb{R}^3$ and $\mathbf{b}_i \in \mathbb{R}^N$. The method works by first splitting the measurement matrix in a rigid and non-rigid part as

$$\mathbf{W} = \mathbf{W}_0 + \delta\mathbf{W} = \mathbf{M}_0\mathbf{B}_0 + \delta\mathbf{M}\delta\mathbf{B}. \tag{4.7}$$

Here, the rigid part $\mathbf{W}_0$ is obtained from the first 3 singular vectors of the measurement matrix where we use the definition.

$$\mathbf{W}_0 = \underbrace{\mathbf{U}_0\mathbf{\Sigma}_0}_{\mathbf{M}_0}\underbrace{\mathbf{V}_0^{\mathrm{T}}}_{\mathbf{B}_0}, \qquad \mathbf{M}_0 \in \mathbb{R}^{2F\times 3}, \qquad \mathbf{B}_0 \in \mathbb{R}^{3\times N}. \tag{4.8}$$

So far, we have shown how to recover the average 3D reconstruction $\mathbf{B}_0$ from a



(a) Corresponding 2D points extracted from StyleGAN images.

(b) 3D reconstruction of the rigid basis shape projected under different rotations.

Figure 4.1: **Illustration of input data and the rigid basis shape.** (a) shows the corresponding 2D points which is used as the input data for the NRSfM algorithm. These 2D points are extracted as landmarks from synthetic StyleGAN images. Only the rightmost column is used in the NRSfM algorithm which received no information about the pixels of the original image other than the coordinates for the extracted 2D landmarks. (b) shows the rigid basis shape $\mathbf{B}_0$ which is determined from the first three right hand singular vectors of the measuring matrix. To illustrate that $\mathbf{B}_0$ is a 3D shape it is projected into the image plane using various rotations.

data set consisting only of corresponding 2D points. In Paper III, we extract such corresponding 2D points from synthetic StyleGAN images using a pretrained landmark extractor. Figure 4.1a shows examples of the synthetic images as well as the extracted 2D landmarks. Figure 4.1b provides an illustration of the recovered rigid 3D basis shape $\mathbf{B}_0$ under various rotations.

In Eq. (4.7), the rigid basis shape $\mathbf{B}_0$ describes the average 3D reconstruction, and $\mathbf{M}_0$ contains the 2×3 projection matrices associated with each of the F observations in the measurement matrix

$$\mathbf{M}_0 = \begin{bmatrix}\mathbf{M}_1 & \mathbf{M}_2 & \cdots & \mathbf{M}_N\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{2F\times3}. \tag{4.9}$$

Each of these individual projection matrices $\mathbf{M}_i$ can be further factorized into

Algorithm 2 Recover camera and rotation matrix from projection matrix

Input: $\mathbf{M} \in \mathbb{R}^{2\times 3}$ ▷ Projection matrix
Output: $\mathbf{K} \in \mathbb{R}^{2\times 2}$, $\mathbf{R} \in \mathbb{R}^{2\times 3}$ ▷ Camera and rotation matrix
$\mathbf{M}' = \text{flipud}(\mathbf{M})$ ▷ Flip up-down
$\mathbf{Q}, \tilde{\mathbf{R}} \leftarrow \text{QR}(\mathbf{M}'^{\text{T}})$ ▷ QR Decomposition
$\mathbf{K}' \leftarrow \text{flipud}(\tilde{\mathbf{R}}^{\text{T}})$
$\mathbf{K}'' \leftarrow \text{fliplr}(\mathbf{K}')$ ▷ Flip left-right
$\mathbf{R}' \leftarrow \text{flipud}(\mathbf{Q}^{\text{T}})$
$\mathbf{S} \leftarrow \text{diag}(\text{sign}(\text{diag}(\mathbf{K}'')))$ ▷ Ensure $\mathbf{K}$ will have positive diagonal
$\mathbf{K} \leftarrow \mathbf{K}''\mathbf{S}$
$\mathbf{R} \leftarrow \mathbf{S}\mathbf{R}'$
return $\mathbf{K}, \mathbf{R}$

upper triangular camera matrices $\mathbf{K}_i$ and rotation matrices $\mathbf{R}_i$ as

$$\mathbf{M}_i = \mathbf{K}_i[\mathbf{I}|\mathbf{0}]\mathbf{R}_i = \begin{bmatrix} k_{11}^{(i)} & k_{12}^{(i)} \\ 0 & k_{22}^{(i)} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11}^{(i)} & r_{12}^{(i)} & r_{13}^{(i)} \\ r_{21}^{(i)} & r_{22}^{(i)} & r_{23}^{(i)} \\ r_{31}^{(i)} & r_{32}^{(i)} & r_{33}^{(i)} \end{bmatrix}. \tag{4.10}$$

The procedure for the decomposition of the projection matrix $\mathbf{M}$ into camera matrix $\mathbf{K}$ and rotation matrix $\mathbf{R}$ as in Eq. (4.10) relies on QR factorization is given in Algorithm 2.

In Paper III, we choose to parameterize the rotation matrices $\mathbf{R}_i$ by Euler angles $\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_z)$. Thus the matrices $\mathbf{R}_i$ are composed of 3-by-3 rotation matrices where we chose the ordering $\mathbf{R}(\boldsymbol{\theta}) = \mathbf{R}_z(\theta_z)\mathbf{R}_y(\theta_y)\mathbf{R}_x(\theta_x)$ where $\mathbf{R}_x$, $\mathbf{R}_y$ and $\mathbf{R}_z$ are the standard rotation matrices in three dimensions given by

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}, \mathbf{R}_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \mathbf{R}_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.11}$$

The non-rigid basis shapes can be defined as follows. First we define the the non-rigid part of the measurement matrix by subtracting the rigid part. We then

calculate the SVD of the resultant matrix

$$\delta\mathbf{W} = \mathbf{W} - \mathbf{W}_0 = \delta\mathbf{U}\delta\mathbf{\Sigma}\delta\mathbf{V}^{\mathrm{T}} = \delta\mathbf{M}\delta\mathbf{B}, \tag{4.12}$$

where we define $\delta\mathbf{B} \equiv \delta\mathbf{V}^{\mathrm{T}} \in \mathbb{R}^{N\times N}$. Now, the main idea of the rank-one approach to NRSfM (Brandt and Ackermann, 2019) is to use the first K rows of $\delta\mathbf{B}$, denoted as $\mathbf{b}_1^{\mathrm{T}}, \mathbf{b}_2^{\mathrm{T}}, \cdots \mathbf{b}_K^{\mathrm{T}} \in \mathbb{R}^{1\times N}$. These are used to construct the non-rigid basis shapes as $\mathbf{B}_k = \mathbf{d}_k\mathbf{b}_k^{\mathrm{T}} \in \mathbb{R}^{3\times N}$ where the vectors $\mathbf{d}_k \in \mathbb{R}^3$ are constrained to have unit norm and found by minimizing the reprojection error

$$\min_{\alpha_{fk},\mathbf{d}_k} \sum_{f=1}^{F} \left\| \mathbf{X}_f - \mathbf{M}_f \left[\mathbf{B}_0 - \sum_{k=1}^{K} \alpha_{fk}\mathbf{d}_k\mathbf{b}_k^{\mathrm{T}} \right] \right\|_F^2 \quad \text{s.t} \quad \mathbf{d}_k^{\mathrm{T}}\mathbf{d}_k = 1. \tag{4.13}$$

Figure 4.2 visualizes the effect of applying each of the first six non-rigid basis shapes $\mathbf{B}_k$ to the rigid basis shape $\mathbf{B}_0$.

4.2 Controlling the 3D geometry of StyleGAN

In Paper III, we propose a method for combining NRSfM with a DGMs like StyleGAN. We first write the NRSfM model described in the previous section in closed form as

$$\mathcal{R}(\mathbf{q}) = \underbrace{\mathbf{K}[\mathbf{I}_2|\mathbf{0}]\mathbf{R}(\boldsymbol{\theta})}_{\mathbf{M}} \left[\mathbf{B}_0 + \sum_{k=1}^{K} \alpha_k\mathbf{B}_k \right] + \mathbf{t} \otimes \mathbf{1}_L^{\mathrm{T}}, \tag{4.14}$$

where $\mathbf{q} = (\mathbf{k}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{t})$ is an *attribute* vector, describing the camera parameters $\boldsymbol{k} = (k_{11}, k_{12}, k_{22})$, rotation $\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_z)$, non-rigid basis shape coefficients $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_K)$ and two dimensional translation $\mathbf{t} = (t_x, t_y)$. Thus, the attribute parameter $\mathbf{q}$ completely specifies the sparse 3D structure via $\boldsymbol{\alpha}$ as well as its projection onto the image plane via $\mathbf{k}$, $\boldsymbol{\theta}$ and $\mathbf{t}$.

The model $\mathcal{R}$ can be seen as a mapping from the space of attribute vectors $\mathbf{q} \in \mathcal{Q}$ to the space of 2D landmarks. In the following our aim is to connect the NRSfM model in Eq. (4.14) with the latent space of StyleGAN. In Paper III, we propose

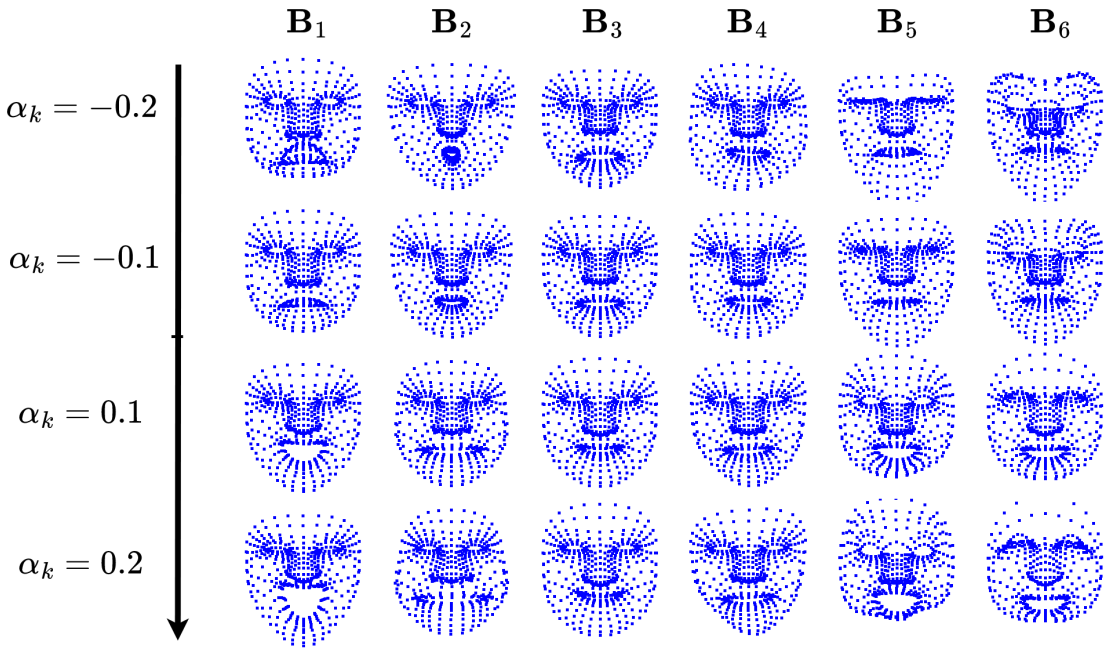


Figure 4.2: **Effect of the non-rigid basis shapes** The figure shows the effect of the non-rigid basis shapes when they are added to the rigid basis shape individually as $\mathbf{B}_0 + \alpha_k \mathbf{B}_k$. The columns corresponds to $k = 1, \cdots, 6$ and the rows show the effect of applying each basis space with the strength parameter α_k increasing from top to bottom.

doing this by training a regressor network ϕ to predict the attribute vector $\mathbf{q}$ of the model in Eq. (4.14) directly from individual StyleGAN latent codes $\mathbf{w}$ such that $\phi(\mathbf{w}) = \mathbf{q}$.

The regressor network ϕ is implemented as a simple MLP network that is trained using a L_2 loss between the predicted 2D landmarks $\mathcal{R}(\phi(\mathbf{w}))$ and the "ground truth" 2D landmark, which we extract from generated StyleGAN images $G(\mathbf{w})$ using a pretrained landmark extractor denoted as ψ_L. The training loss is then written as

$$\mathcal{L}(\mathbf{w}) = \|\mathcal{R}(\phi(\mathbf{w})) - \psi_L(G(\mathbf{w}))\|_F^2 . \tag{4.15}$$

We propose two methods for using the trained regressor model ϕ to control the images generated by StyleGAN, a linear method and an iterative gradient-based method. The linear method is derived from the first order Taylor expansion of ϕ which can be rearranged to obtain a formula describing how the latent code should

be updated to achieve a specified edit

$$\phi(\mathbf{w}) \approx \phi(\mathbf{w}_0) + \mathbf{J}|_{\mathbf{w}=\mathbf{w}_0}(\mathbf{w} - \mathbf{w}_0) \rightarrow \mathbf{w} = \mathbf{w}_0 + \mathbf{J}^{\dagger}(\mathbf{q} - \mathbf{q}_0), \tag{4.16}$$

where $\phi(\mathbf{w}) = \mathbf{q}$ and $\phi(\mathbf{w}_0) = \mathbf{q}_0$ and $\mathbf{J}^{\dagger}$ is the Moore-Penrose (Penrose, 1955) pseudo-inverse of the Jacobian of ϕ evaluated at $\mathbf{w}_0$. The linear method is attractive as it gives a closed form solution for semantic editing with just a single step. Because editing only requires a single application of Eq. (4.16), the linear method is fast and can run in near-real time on a consumer-grade GPU.

As an alternative to the linear method in Eq. (4.16), we further propose a gradient-based method for semantic editing, that can be formulated as

$$\min_{\mathbf{w}} \|\phi(\mathbf{w}) - \mathbf{q}_{\text{target}}\|_2^2 + \lambda \mathcal{D}(G(\mathbf{w}), G(\mathbf{w}_0)), \tag{4.17}$$

where the $\mathcal{D}$ is an image similarity metric that we add as a regularization term with a strength parameter $\lambda \in \mathbb{R}^+$. In Paper III, we show that although the linear method is able to edit latent codes such that structure of the edited images agrees with the specified target attribute vector $\mathbf{q}$, the method suffers from a shift in identity during edits. The gradient-based method, although slower, can effectively increase the degree of identity preservation by using a pretrained ArcFace network (Deng et al., 2019) as a the regularization term.

Figure 4.3 shows the effectiveness of this approach, by conditioning the sampling on different attribute vectors. Without conditioning, the generated faces have a variety of poses and facial expressions as shown in Figure 4.3a. The approach proposed in Paper III allows for conditional sampling by specifying the geometry by choosing a particular desired attribute vector. Figure 4.3b, 4.3c and 4.3d show conditional sampling for attribute vectors corresponding to a neutral frontal view, a smiling expressions and a rotated view respectively.

(a) Unconditional sampling (b) Neutral frontal view.

(c) Smiling. (d) Rotated view.

Figure 4.3: **Controllable sampling.** Our method allows for controllable synthesis by conditioning on a specific attribute vector $\mathbf{q}$. Using unconditional sampling as shown in (a) the generated faces have a variety of poses and facial expressions. By choosing an appropriate attribute vector, we can sample the same identities from a frontal view with a neutral expression as shown in (b) or with a smile or specific pose as shown in (c) and (d), respectively.

4.3 Conclusions

The tensor model from Paper I and Paper II ultimately finds linear editing directions each corresponding to a specific semantic change in the output image. Thus we end up with a collections of seven directions $\mathbf{n} \in \mathcal{W}+$ corresponding to the six prototypical facial expressions as well as a direction for yaw rotation. While the tensor model is only able to control yaw rotation, the NRSfM approach in Paper III is able to define arbitrary poses. When combined, these two approaches allow for controlling the facial expressions as well as poses of face images generated

with StyleGAN. While the tensor model is only able to find linear directions, the NRSfM approach is able find both linear editing directions with Eq. (4.16) as well as non-linear editing trajectories using Eq. (4.17).

As mentioned in Section 2.5, StyleRIG (Tewari et al., 2020) aims to gain more explicit control over the synthesis process in StyleGAN by using a 3DMM. In comparison, the method proposed in Paper III does not require access to a pretrained 3DMM and relies only on access to a pretrained landmark extractor on the domain of the generator. Since StyleRIG renders synthetic images from the 3DMM, they are able to control the illumination of the generated images. Our method is not able to model illumination. Our method shares limitation with StyleRIG as modes of variation, such as background or hair style, that are not modeled by either our NRSfM model or the 3DMM from StyleRIG cannot be explicitly controlled.

In order to achieve greater 3D controllability Gu et al. (2022) proposed to integrate Neural Radiance Fields (NeRFs) (Mildenhall et al., 2020) into the StyleGAN architecture. Compared to our approach in Paper III, StyleNeRF requires both adaptation to the StyleGAN architecture as well as expensive retraining, whereas our method can be applied to existing model checkpoints without the need for altering the StyleGAN architecture or any retraining of the generator.

The editing techniques proposed in Paper III are in some way related to the methods proposed in Hijack-GAN by Wang et al. (2021) which was introduced in Section 2.5. In Paper III, we train an regressor network ϕ to predict an attribute vector that encodes information about the 3D geometry and orientation of the generated faces. In our work, the regressor ϕ plays a similar role to the proxy network from Hijack-GAN. However, as we do not frame the editing process as an attack problem there is no restriction with respect to access of the gradients of the generator. We leverage this access to the gradients to apply identity regularization which improves identity preservation along the editing trajectories. Further having explicit access to the generator enables us to work in the intermediate $\mathcal{W}$ space allowing for more disentangled edits.

Chapter 5

An interactive art experience

This chapter introduces Paper IV: The paper takes a departure from the focus on semantic editing of face images and instead explores another application of StyleGAN. The paper was a collaboration with a fellow PhD student and was submitted as an extended abstract to CHI2023.

Some museums have accumulated vast amounts of digitized artworks that can contain thousands of individual pieces of art. In collaboration with the Munch Museum in Oslo (MUNCH), we had access to a digital collection of 5800 crayon, ink, and pencil drawings made by the famous artist Edvard Munch. It is a challenging task to showcase such vast collections adequately using conventional museum exhibitions. In Paper IV, we propose to utilize a StyleGAN model in conjunction with a pSp encoder (Richardson et al., 2021) to design an interactive experience consisting of a drawing table where the museum audience can directly interact with the trained models. This allowed the users to explore the sketching style of Edward Munch though the representations learned by these models.

We first trained a StyleGAN2 model on the Edward Munch collection. This allowed us to generate new artworks that follow the particular style of the collection. Figure 5.1a shows samples from the trained StyleGAN model along with interpolation results in Figure 5.1b.

After training StyleGAN on the data set, the next step was to train a pSp encoder

(a) Samples from the model. (b) Interpolation in latent space.

Figure 5.1: **StyleGAN2 trained on a collection of artworks by Edvard Munch**. (a) shows random samples from the trained model. (b) shows interpolations in $\mathcal{W}$ space of the trained model.

(Richardson et al., 2021). The architecture of the pSp encoder was introduced in Section 2.4.

To facilitate inversion from a sketch provided by the user to the domain of the data, we first simplified images generated with the StyleGAN model such that they are closer to the expected input from the user. We do this as proposed by Richardson et al. (2021) by first applying a "pencil sketch" filter[1] to the generated images and then simplifying the result using the pre-trained deep "sketch-simplification" model provided by Simo-Serra et al. (2016).

We sample 10K synthetic images $\mathbf{x}$ from the trained StyleGAN model and for each image, we obtain a corresponding simplified sketch $\widetilde{\mathbf{x}}$ which we can use to train the pSp encoder E. The encoder is trained to predict the offset from the average latent code $\widehat{\mathbf{w}}$ of the trained StyleGAN generator. Thus, the pSp model is defined as $\text{pSp}(\mathbf{x}) := G(E(\mathbf{x}) + \bar{\mathbf{w}})$ (Richardson et al., 2021).

In order to facilitate domain translation from the simplified input sketches, we train the pSp encoder to make reconstructions based on the simplified sketches $\widetilde{\mathbf{x}}$ as $\widehat{\mathbf{x}} = \text{pSp}(\widetilde{\mathbf{x}})$ such that the reconstruction $\widehat{\mathbf{x}}$ are as close to the original $\mathbf{x}$ as possible. This is done by minimizing the L_2 and LPIPS distance between the generated synthetic samples and the reconstructions $\widehat{\mathbf{x}}$. As in the work by Richardson et al.

[1]The code for this pencil "sketch filter" is provided by Richardson et al. at github.com/eladrich/pixel2style2pixel/blob/master/scripts/generate_sketch_data.py

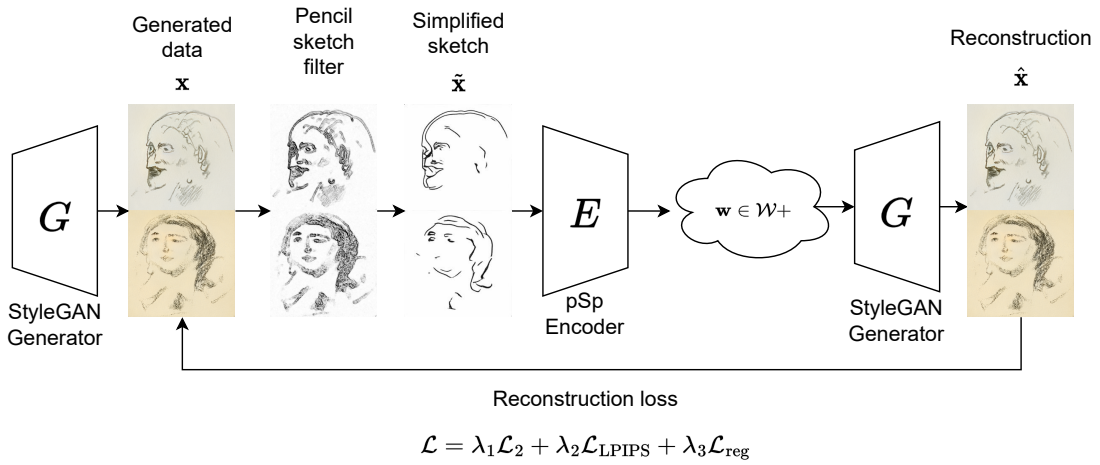


Figure 5.2: **Sketch generation pipeline**. Generated images from the trained StyleGAN model are first processed to generate simplified sketches. We then train the pSp encoder to map the simplified sketches into the $\mathcal{W}+$ space of the StyleGAN model.

(2021) we also employ a regularization loss $\mathcal{L}_{\text{reg}}(\tilde{\mathbf{x}}) = ||E(\tilde{\mathbf{x}}) - \bar{\mathbf{w}}||_2$ that encourages the output of the encoder to stay close to the average latent code $\bar{\mathbf{w}}$. An illustration of the training pipeline is shown in Figure 5.2.

This research project culminated in the creation of an interactive drawing table where the user draws with a pen on a piece of tracing paper. A camera under the table continuously captures the current state of the user-generated sketch. The sketch images are converted to binary and the frames are then sent to the pSp model that projects each frame into the $\mathcal{W}+$ space of the trained StyleGAN model. New images are then synthesized from the inverted latent codes and are projected back onto the drawing table in order to provide feedback to the user. This setup allows the user to directly interact with the trained StyleGAN model and by extension interact with the collection of Edvard Munch artworks that the model was trained on. This provides a new way of interacting with the art collection, which would be too large to adequately show to the audience otherwise.

Chapter 6

Denoising Diffusion Models

This chapter serves to introduce Paper V. Contrary to the other contributions within this thesis, Paper V shifts the focus from GANs to explore novel methods for semantic editing in an alternative class of generative models, namely, Denoising Diffusion Models (DDMs) (Sohl-Dickstein et al., 2015). DDMs have recently emerged as a powerful class of generative models with remarkable capabilities in producing high-quality images from diverse domains. In line with the topic of this thesis, Paper V focuses on the use of DDMs for controllable generation of face images.

Section 6.1 will introduce the reader to the core ideas of DDMs focusing on the framework of Denoising Diffusion Probabilistic Modelss (DDPMs) as proposed by Ho et al. (2020) and the deterministic reformulation proposed by Song et al. (2021) known as Denoising Diffusion Implicit Models (DDIMs).

Section 6.2 introduces the notion of the *semantic latent space* within DDMs as proposed by Kwon et al. (2023) and will summarize the contributions of Paper V that proposes novel editing techniques in DDMs by utilizing the semantic latent space.

6.1 Diffusion Models

DDMs are a new class of deep generative models that have recently emerged as a strong competitor to GANs and even surpassing them (Dhariwal and Nichol, 2021) on unconditional image synthesis where GANs have otherwise had a dominating role in the field. As DDMs are are a fairly new type of generative models, the rest of this section serves as an introduction and will review the core theory of DDPMs and DDIMs.

On a high level, the aim is DDMs is to approximate the true distribution of the training data $q(\mathbf{x}_0)$ by some learned distribution $p_\theta(\mathbf{x}_0)$ such that we can sample new data points $\mathbf{x}_0 \sim p_\theta(\mathbf{x}_0)$. Intuitively, DDMs are characterized by two distinct processes, a forward process, and a reverse process. The forward process gradually adds Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to data $\mathbf{x}_0$ in T steps, starting from a clean image $\mathbf{x}_0$ and producing a sequence of progressively more noisy images $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{T-1}, \mathbf{x}_T$. In the generative reverse process, we would like to move in the other direction. Given a fully noised sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we aim to learn a model that can gradually remove the noise and give us a new sample $\mathbf{x}_0 \sim p_\theta(\mathbf{x}_0)$. The generative process of DDMs is illustrated in Figure 6.1 for a model trained on face images.

In DDPM (Ho et al., 2020), the forward noising process is defined as a fixed Markov chain where Gaussian noise is gradually added to the training data according to a variance schedule $\beta_1, \beta_2, \cdots, \beta_T$ as

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \tag{6.1}$$

Using the reparameterization trick for Gaussians, we can sample create a sample $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_{t-1})$ as

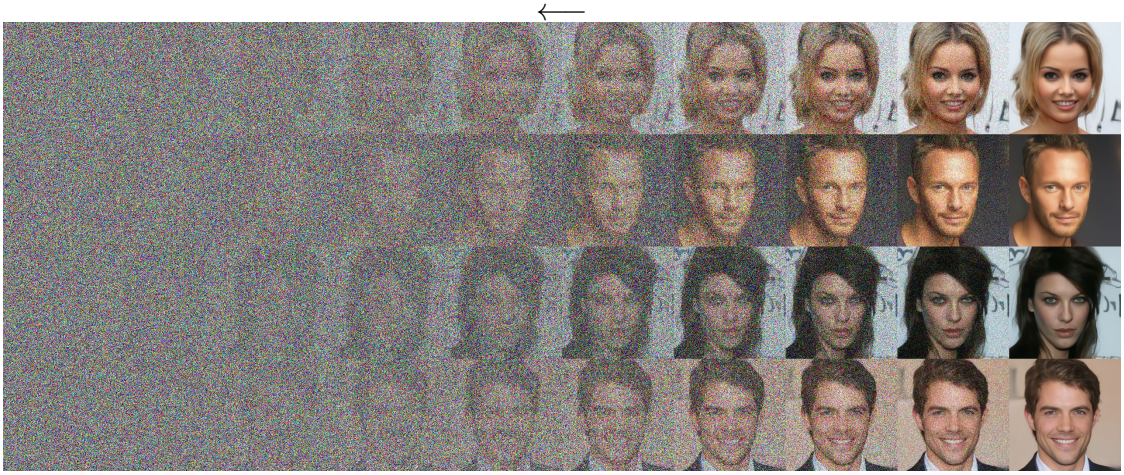
$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon} \quad \text{with} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{6.2}$$

An interesting property of the forward process in Eq. (6.1) is that it allows for sampling at an arbitrary noise level in a single step directly from the clean image

Forward noising process

$\mathbf{x}_{t+1} \sim q(\mathbf{x}_{t+1}|\mathbf{x}_t)$

$\longleftarrow$



$\longrightarrow$

Reverse generative denoising process

$\mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$

Figure 6.1: **Illustration of the generative process in DDMs.** The denoising process of DDMs is illustrated by showing the variable $\mathbf{x}_t$ as various timesteps during the generative process.

$\mathbf{x}_0$. By defining[1] $\alpha_t \equiv 1 - \beta_t$ and $\bar{\alpha}_t \equiv \prod_{i=1}^{t} \alpha_i$, we can write

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{I}) \tag{6.3}$$

This property can be shown as follows. First, recall that a Gaussian random variable $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I})$ can be sampled as $\mathbf{x} = \boldsymbol{\mu} + \sigma\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. To prove Eq. (6.3), we first sample from Eq. (6.1) as

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{1 - \beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}_{t-1} && (6.4)\\
&\equiv \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1 - \alpha_t}\boldsymbol{\epsilon}_{t-1} && (6.5)\\
&= \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1 - \alpha_{t-1}}\boldsymbol{\epsilon}_{t-2}) + \sqrt{1 - \alpha_t}\boldsymbol{\epsilon}_{t-1} && (6.6)\\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t(1 - \alpha_{t-1})}\boldsymbol{\epsilon}_{t-2} + \sqrt{1 - \alpha_t}\boldsymbol{\epsilon}_{t-1} && (6.7)
\end{aligned}$$

[1]Note that in the DDIM paper (Song et al., 2021) and many other works, the symbol α_t is defined as the cumulative product, which is denoted as $\bar{\alpha}_t$ in this text following Ho et al. (2020).

Now, the two last terms can be rewritten by merging two Gaussians (*i.e.*, if $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \sigma_1^2\mathbf{I})$ and $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \sigma_2^2\mathbf{I})$ then $\mathbf{x} + \mathbf{y} \sim \mathcal{N}(\mathbf{0}, (\sigma_1^2 + \sigma_2^2)\mathbf{I})$), so

$$\alpha_t(1 - \alpha_{t-1}) + (1 - \alpha_t) = 1 - \alpha_t\alpha_{t-1}. \tag{6.8}$$

We can merge the two Gaussian variables $\boldsymbol{\epsilon}_{t-2} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\boldsymbol{\epsilon}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ into a combined Gaussian variable $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and continue the calculation as

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1 - \alpha_t\alpha_{t-1}}\boldsymbol{\epsilon} && (6.9)\\
&= \sqrt{\alpha_t\alpha_{t-1}\alpha_{t-2}}\mathbf{x}_{t-3} + \sqrt{1 - \alpha_t\alpha_{t-1}\alpha_{t-2}}\boldsymbol{\epsilon} && (6.10)\\
&= \sqrt{\prod_{i=1}^{T} \alpha_i}\mathbf{x}_0 + \sqrt{1 - \prod_{i=1}^{T} \alpha_i}\boldsymbol{\epsilon} && (6.11)\\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, && (6.12)
\end{aligned}$$

which proves the form of $q(\mathbf{x}_t|\mathbf{x}_0)$ in Eq. (6.3). Thus we have shown that at every timestep t each noise image $\mathbf{x}_t$ can be obtained as a linear combination of the original clean image $\mathbf{x}_0$ and Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ as

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon} \tag{6.13}$$

It is worth noting that the noise schedule is chosen such that $\bar{\alpha}_T \approx 0$ in order to ensure that $q(\mathbf{x}_T|\mathbf{x}_0) \approx \mathcal{N}(\mathbf{x}_T, \mathbf{0}, \mathbf{I})$.

Having established how noise can be efficiently added to the training data to obtain progressively noisier samples using the forward process, we now transition to task of learning a model that can remove the noise. The objective is to approximate the reverse conditional distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ by employing a neural network parameterized by θ, denoted as $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$. Approximating the true reverse conditional distribution enables us to sample $\mathbf{x}_T \sim p(\mathbf{x}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ and subsequently use $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ in a sequential manner to reverse the noising process and ultimately obtain a novel sample following the data distribution $\mathbf{x}_0 \sim q(\mathbf{x}_0)$.

The true reverse distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is intractable since it would require knowledge of the entire data set (Ho et al., 2020). However, it is noteworthy that it

becomes tractable when conditioned on $\mathbf{x}_0$ as $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$. This makes intuitive sense since if we know the clean image and a noisy version of it should be possible to approximate the intermediate steps. Using Bayes Theorem, we can write the tractable posterior as

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}. \tag{6.14}$$

Now, due to the Markov property of the forward process in Eq. (6.1), we have that $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) = q(\mathbf{x}_t|\mathbf{x}_{t-1})$. Further, the expression for $q(\mathbf{x}_{t-1}|\mathbf{x}_0)$ can be derived directly from the expression for $q(\mathbf{x}_t|\mathbf{x}_0)$ in Eq. (6.13). Thus, all terms in Eq. (6.14) are known and can be written as explicit closed form Gaussians. These can be plugged into Eq. (6.14) and the expression simplified. In practice, this results in a fairly involved calculation of which Luo has provided a very detailed write-up (Luo, 2022, p.12). The result of the calculation is that the tractable posterior $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ yields a new Gaussian

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0, t), \tilde{\beta}_t) \tag{6.15}$$

where the true denoising mean $\tilde{\boldsymbol{\mu}}$ is given by

$$\tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0, t) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \tag{6.16}$$

and true forward process variance $\tilde{\beta}_t$ is given by

$$\tilde{\beta} = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \tag{6.17}$$

Now, using the property in Eq. (6.13), we can write $\mathbf{x}_0$ in terms of $\mathbf{x}_t$ as

$$\mathbf{x}_0 = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\boldsymbol{\epsilon}_0. \tag{6.18}$$

Plugging this back into Eq. (6.16) we can write the true transition mean of the

forward process as

$$\tilde{\boldsymbol{\mu}}(\mathbf{x}_t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t + \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}}}\boldsymbol{\epsilon}_0\right). \tag{6.19}$$

We can parameterize the reverse processes as a neural network $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ as

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)), \tag{6.20}$$

where Eq. (6.19) shows that we can model the approximate transition mean $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ in terms of a neural network $\boldsymbol{\epsilon}_\theta$ that is trained to predict the noise $\boldsymbol{\epsilon}$ at timestep t

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t + \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right). \tag{6.21}$$

Ho et al. (2020) proposed to set the forward variance to untrained time dependent constants $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) = \sigma_t^2\mathbf{I}$ where the choices $\sigma_t^2 = \tilde{\beta}_t$ and $\sigma_t^2 = \beta_t$ were reported to produce similar experimental results. In general, there is some freedom of design in how the forward variances β_t are set. Ho et al. (2020) proposed to define the variance schedule as a sequence of linearly increasing constants, starting from $\beta_1 = 10^{-4}$ all the way up to $\beta_T = 0.02$. In follow-up work by Nichol and Dhariwal (2021), the authors noted that although the linear noise schedule proposed by Ho et al. (2020) worked well for high-resolution images, it was sub-optimal for images of resolutions 64×64 and 32×32. To address this problem Nichol and Dhariwal (2021) proposed a cosine noise schedule defined as

$$\bar{\alpha}_t = \frac{f(t)}{f(0)} \quad \text{with} \quad f(t) = \cos^2\left(\frac{t/T + s}{1+s}\frac{\pi}{2}\right), \tag{6.22}$$

where s is a small offset.

With $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ parameterized as in Eq. (6.21), Ho et al. (2020) proposed a simplified training objective that minimizes the squared L_2 distance between $\boldsymbol{\epsilon}$ and $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$

as

$$L_{\text{simple}} = \mathbb{E}_{t\sim U(0,T),\mathbf{x}_0\sim q(\mathbf{x}_0),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}}\boldsymbol{\epsilon}}_{\mathbf{x}_t}, t)\right\|^2\right]. \tag{6.23}$$

After training the network $\boldsymbol{\epsilon}_\theta$ to predict the added noise, we can use it to generate new samples. The DDPM sampling procedure starts by drawing a Gaussian sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and then iteratively applying the denoiser as

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t + \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{6.24}$$

So far, this text has treated DDMs in the context of the DDPM framework provided by Ho et al. (2020). Although DDPMs are able to create images with impressive quality and variation, they require many function evaluations to produce a single sample. Typically DDPMs are trained with a thousand timesteps in the forward process and require the same amount of function evaluations to run the reverse process. In contrast, GANs only need a single forward pass though the generator to produce a sample. The need to make many sequential function evaluations makes DDPMs much slower than GANs. Comparing the two generative architectures, Song et al. (2021) noted that it would take around 20 hours to sample 50k images of size 32×32 on a Nvidia 2080 Ti GPU using a DDPMs, where sampling the same amount of images using a GANs would take less than a minute.

In order to speed up the sampling of DDMs, Song et al. (2021) proposed to generalize DDPMs to a larger family of generative process indexed by σ as

$$q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t_1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2\mathbf{I}). \tag{6.25}$$

The idea is that the entire family of models in Eq. (6.25) can be optimized by the same objective function as DDPMs in Eq. (6.23). Thus, a model that is trained for the original DDPM process can be used for any sampling process in this extended family of generative processes.

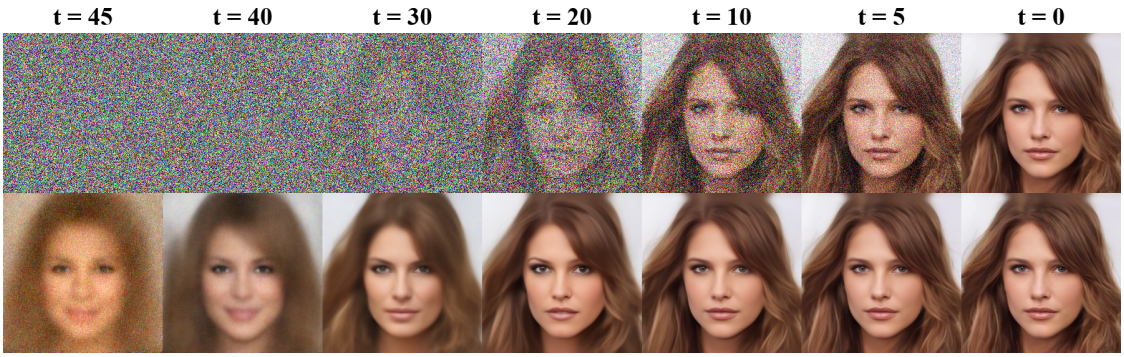


Figure 6.2: **Comparison $\mathbf{x}_t$ and $\mathbf{P}_t(\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t))$ at different timesteps**

The sampling process using DDIM sampling is written as

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \underbrace{\left(\frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t)}{\sqrt{\bar{\alpha}_t}} \right)}_{\text{predicted } \mathbf{x}_0} + \underbrace{\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t)}_{\text{direction pointing to } \mathbf{x}_t} + \underbrace{\sigma_t \boldsymbol{\epsilon}_t}_{\text{noise}} \tag{6.26}$$

with $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and

$$\sigma_t = \eta_t \sqrt{(1-\bar{\alpha}_{t-1})(1-\bar{\alpha}_t)} \sqrt{(1-\bar{\alpha}_t/\bar{\alpha}_{t-1})} = \eta_t \sqrt{\tilde{\beta}} \tag{6.27}$$

When $\eta_t = 1$ for all t in Eq. (6.27) we have that $\sigma_t^2 = \tilde{\beta}$ and the sampling procedure in Eq. (6.26) reduces to the DDPM sampling in Eq. (6.24). When $\eta_t = 0$ for all t, the sampling in Eq. (6.27) is DDIM and the reverse process becomes fully deterministic and reversible. For $0 < \eta_t < 1$ the sampling in Eq. (6.27) the control parameter η_t allows for controlling the level of stochasticity in the generative process. In this way, DDIMs generalizes DDPMs to a larger class of models.

Using the notation proposec by Kwon et al. (2023) and used in Paper V, the prediction of $\mathbf{x}_0$ given $\mathbf{x}_t$ can be written as

$$\mathbf{P}_t(\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t)) = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t)}{\sqrt{\bar{\alpha}_t}}. \tag{6.28}$$

During the sampling process, the prediction $\mathbf{P}_t(\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t))$ "clears up" much faster than he latent variables $\mathbf{x}_t$, which is illustrated in Figure 6.2. In the DDIM setting

($\eta_t = 0$) the deterministic generative process can then be written as

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\mathbf{P}_t(\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t)) + \sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t). \tag{6.29}$$

It is possible to reverse Eq. (6.29) such as to uniquely determine the particular $\mathbf{x}_T$ that would produce a given clean image $\mathbf{x}_0$. In the limit of small steps, the inversion can be written as

$$\mathbf{x}_{t+1} = \sqrt{\bar{\alpha}_{t+1}}\mathbf{P}_t(\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t)) + \sqrt{1-\bar{\alpha}_{t+1}}\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t). \tag{6.30}$$

The DDIM framework allows for much faster inference than DDPM by sampling only a subset of the latent variables during the generative reverse process. Instead of sampling all $\mathbf{x}_t$ for $t \in [1, \cdots, T]$, Song et al. (2021) proposed to only sample a subset of the latent variables $\{\mathbf{x}_{\tau_1}, \cdots, \mathbf{x}_{\tau_i}, \cdots, \mathbf{x}_{\tau_S}\}$ with $S < T$ during the reverse process. This results in a significant speedup as good quality samples can be achieved with far fewer evaluations of $\boldsymbol{\epsilon}_\theta$.

6.2 Editing in the semantic latent space.

While there exists extensive literature dealing with semantic editing in the latent space of GANs, a selection of which has been treated in Section 2.5, the same is not true in the case of unconditional DDMs. The ease of applying text-based conditioning in DDMs using Classifier-Free Guidance (CFG) (Ho and Salimans, 2021) has resulted in a surge of works dealing with text-based synthesis and editing using DDMs, see *e.g.* Ramesh et al. (2022), Saharia et al. (2022), Kawar et al. (2023), Ruiz et al. (2023), and Gal et al. (2022).

As noted by Luo (2022), it is currently understood that a main drawback of DDMs when compared to other types of generative models such as GANs, is that they do not produce interpretable latents since the intermediate latent variables in the diffusion chain are restricted as noisy versions of the original input. Further, since the latent variables are restricted to have the same dimensionality as the input, DDMs do not enable a compressed latent representation that carries semantic information.

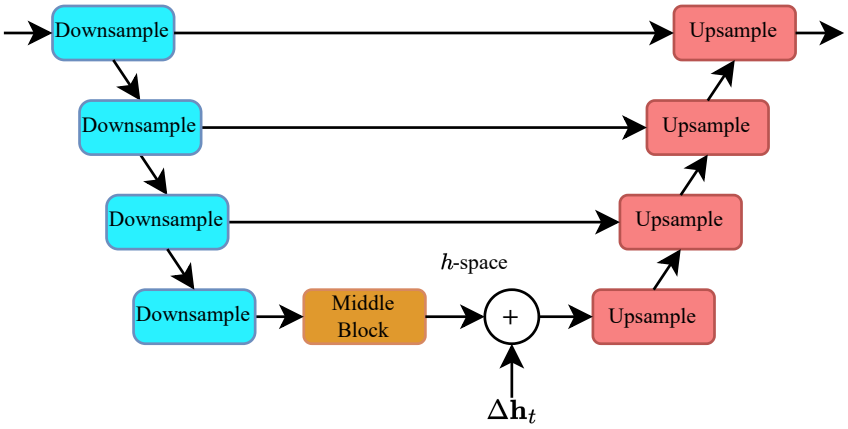


Figure 6.3: **Illustration of the U-Net architecture** The U-Net is a model architecture that produces an output of the same dimensionality as the input. The semantic latent space in diffusion models consists of the space of activations of the deepest bottleneck layer of the U-Net. This is denoted by $\Delta \mathbf{h}_t$ in the figure.

The introduction to DDMs in Section 6.1 did not explicitly address the actual implementation or architecture of the neural network suitable to act as the denoiser $\boldsymbol{\epsilon}_\theta$. The DDM formalism introduced in Section 6.1 does not put any restrictions on the type of network that can be used to implement $\boldsymbol{\epsilon}_\theta$ as long as the model represents a mapping $\mathbb{R}^n \to \mathbb{R}^n$, *i.e.*, the input and output dimensions remain the same. Thus, in principle, any architecture is applicable as long as it obeys this constraint.

In practice, most DDM denoisers are implemented as U-Nets (Ronneberger et al., 2015). U-Nets are a type of CNN that was initially developed for semantic segmentation of biomedical images. The architecture consists of an encoder and decoder part with skip connections. The encoder gradually down-samples the input image using convolutional blocks where in each pass, the spatial dimension of the image is decreased while the number of channels is increased.

The semantic latent space of DDMs. Recently, Kwon et al. (2023) introduced the idea of a *semantic* latent space in DDMs. Rather than treating the variables $\{\mathbf{x}_t\}_{t=1}^T$ in the diffusion process as the latent variables, Kwon et al. (2023) suggested looking closer at the intermediate representations in the denoising U-Net. The

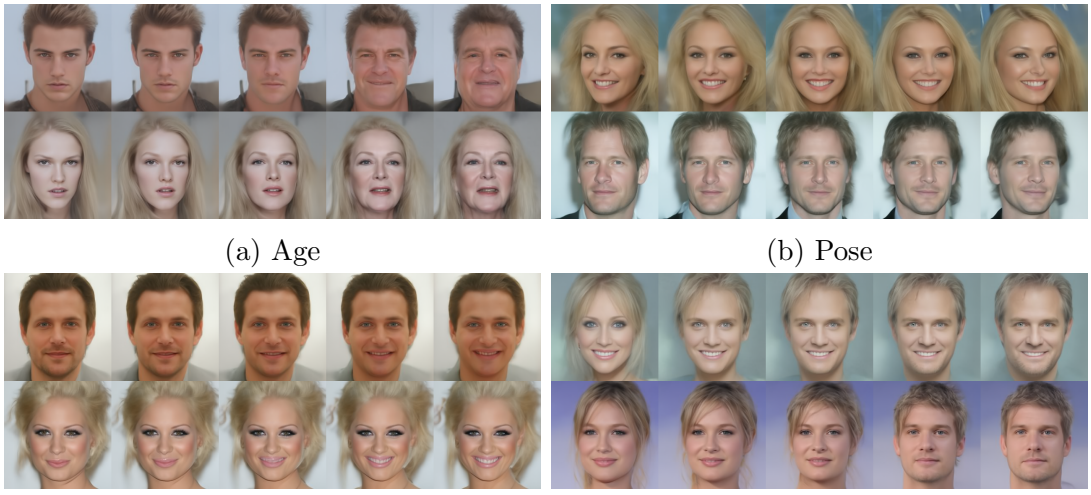
main idea is to define the intermediate representation in the deepest feature map of the denoising U-Net as the latent representation. Kwon et al. (2023) proposed to denote this space as h-space. In each step of the diffusion process, the current noisy image $\mathbf{x}_t$ is fed to the U-Net where it is down-sampled until the deepest feature map $\mathbf{h}_t$. For the model used in Paper V, $\mathbf{x}_t$ has dimensions $(3, 256, 256)$ and $\mathbf{h}_t$ has dimensions $(512, 8, 8)$. A diagram of the U-Net architecture along with an illustration of the semantic h-space is shown in Figure 6.3.

In Paper V, we propose several novel editing techniques for discovering interpretable semantic editing direction in DDMs. To this end, we use the notion of the semantic latent space which is comprised of the bottleneck representations in the U-Net. Since the editing techniques depend on this smaller representation, the methods proposed in Paper V are specific to DDMs which use a U-Net type architecture and are not immediately applicable to DDMs which use other architectures for implementing the denoiser $\boldsymbol{\epsilon}_\theta$.

Principal directions in h-space. As mentioned in Section 2.5, PCA has previously been used to identify semantically meaningful direction in GANs. In GANSpace (Härkönen et al., 2020), PCA directions were shown to lead to interpretable direction in StyleGAN. In Paper V, we perform PCA on a collection of sampled bottleneck representations $\{\mathbf{h}_t\}_{i=1}^N$ and show that many of these directions lead to semantically meaningful editing directions in the semantic latent space.

Concurrent work by Park et al. (2023) also attempts to discover unsupervised directions by working h-space. In their work, due to the high dimensionality of the bottleneck feature maps, the authors proposed using a reduced h-space, consisting of sum-pooled feature maps of the bottleneck representation. In contrast, our methods always work on the full bottleneck representations of the U-net, *i.e.* the full h-space, as proposed by Kwon et al. (2023).

Park et al. (2023) also attempted to find directions in DDMs using PCA, however with limited success. The authors note that PCA directions resulted in severe distortions in the generated images while only somewhat altering attributes such as expression, rotation, and age. The exact reason for the difference in results is unclear, but one might hypothesize that the difference in the definition of the h-



(a) Age (b) Pose

(c) Smile (d) Gender

Figure 6.4: **Interpolation results for semantic direction found using PCA.** In Paper V, we find that many of the latent directions found using PCA carry interpretable semantic meaning such as pose, age, gender, and smile. The found directions allow for smooth interpolation along the different directions in the semantic latent space of DDMs.

space plays an important role. In contrast to the findings of Park et al. (2023), the results of Paper V suggest that directions in h-space, determined using PCA, can modify semantically meaningful attributes of the generated images. In Figure 6.4, this is illustrated by interpolating along directions corresponding to pose, age, gender, and smile.

Image-specific directions. In Paper V, we propose a novel unsupervised technique to discover interesting semantic directions from only a single image. This is in contrast to the PCA method where many samples are needed in order to calculate the principal directions. The intuition is that we seek to find a set of orthogonal directions in h-space that produce the largest change in the image at every timestep during the generative process. In Paper V, we identify such directions as the right hand singular vectors of the Jacobian of the denoiser with respect to h-space at each timestep. The Jacobian at timestep t can be written as $\mathbf{J}_t = \partial \boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t, \mathbf{h}_t)/\partial \mathbf{h}_t$. In practice, the high dimensionality of the h-space quickly becomes a barrier to calculating the Jacobian directly. Rather than redefining the h-space by reducing

the dimensionality by pooling as proposed by Park et al. (2023), we propose finding the top singular vectors of the Jacobian using the power iteration method, which allows us to find the dominant eigenvalues and eigenvectors of $\mathbf{J}_t^{\mathrm{T}}\mathbf{J}_t$. The eigenvectors of $\mathbf{J}_t^{\mathrm{T}}\mathbf{J}_t$ are equivalent to the right-hand singular vectors of the Jacobian $\mathbf{J}_t$. The power iteration method provides a way to find the dominant eigenvector and eigenvalue of a matrix $\mathbf{A}$ by first randomly initializing a non-zero $\mathbf{v}_0$ and then iterating over the recursive formula $\mathbf{v}_{i+1} = \mathbf{A}\mathbf{v}_i/||\mathbf{A}\mathbf{v}_i||$. The vector $\mathbf{v}$ will converge towards the dominant eigenvector of $\mathbf{A}$ with eigenvalue λ given by $\lambda \approx \mathbf{v}^{\mathrm{T}}\mathbf{A}\mathbf{v}/\mathbf{v}^{\mathrm{T}}\mathbf{v}$.

The dominant eigenvectors of $\mathbf{J}_t^{\mathrm{T}}\mathbf{J}_t$ can be calculated without ever explicitly calculating the Jacobian using the following trick. We first calculate the Jacobian Vector Product (JVP) $\mathbf{J}_t\mathbf{v}$ using

$$\mathbf{J}_t\mathbf{v} = \frac{\partial}{\partial a}\boldsymbol{\epsilon}_t^{\theta}(\mathbf{x}_t, \mathbf{h}_t + a\mathbf{v})\bigg|_{a=0}. \tag{6.31}$$

and then calculate an iteration of $\mathbf{J}_t^{\mathrm{T}}\mathbf{J}_t\mathbf{v}$ using

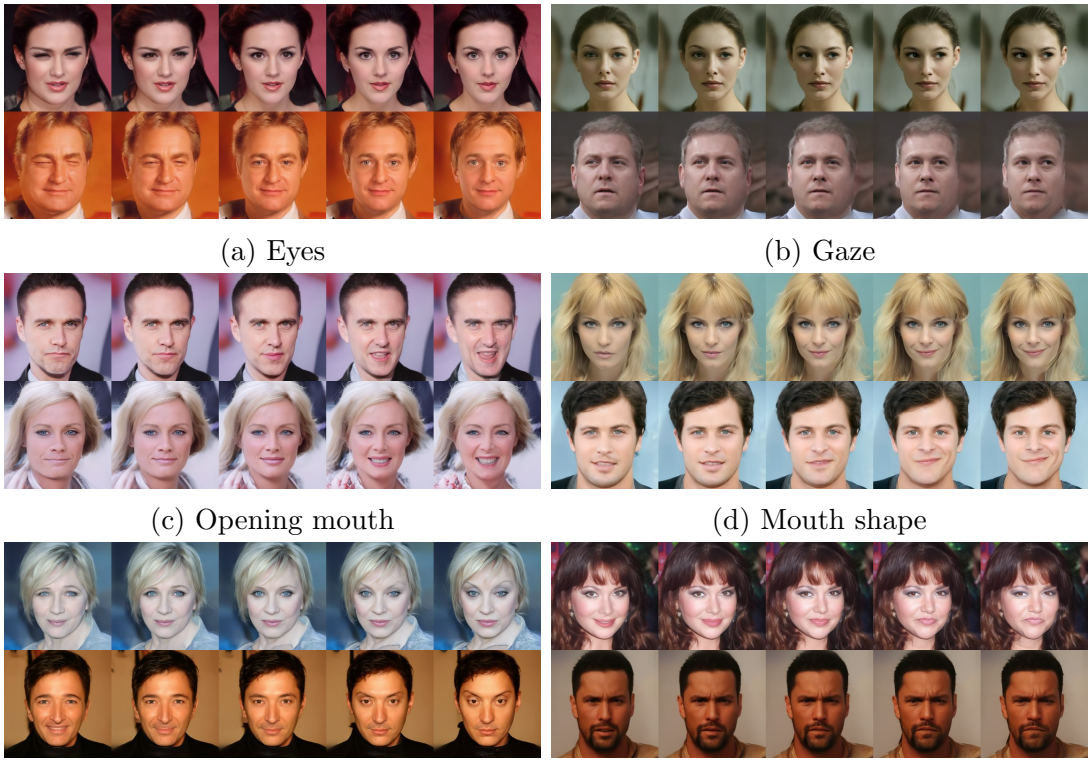
$$\mathbf{J}_t^{\mathrm{T}}\mathbf{J}_t\mathbf{v} = \frac{\partial}{\partial \mathbf{h}_t}\left\langle \boldsymbol{\epsilon}_t^{\theta}(\mathbf{x}_t, \mathbf{h}_t), \mathbf{J}_t\mathbf{v}\right\rangle. \tag{6.32}$$

This process process is then repeated until convergence. This trick is applicable in general and not only in this specific context. Eq. (6.31) is an application of the chain rule. To see this, consider a vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$. The Jacobian of f is denoted as $\mathbf{J} \in \mathbb{R}^{m\times n}$ with components $\mathbf{J}_{ij} = \partial f_i/\partial x_j$. Now, let $\mathbf{u} = \mathbf{x} + a\mathbf{v}$ for some scalar a and constant vector $\mathbf{v} \in \mathbb{R}^n$, then we have component wise

$$\left(\frac{d}{da}f(\mathbf{x} + a\mathbf{v})\right)_i = \left(\frac{d}{da}f(\mathbf{u})\right)_i = \sum_{j=1}^{n}\frac{\partial f_i}{\partial u_j}\frac{du_j}{da} = \sum_{j=1}^{n}\frac{\partial f_i}{\partial u_j}v_j \tag{6.33}$$

which is exactly the ith component of $\mathbf{J}\mathbf{v}$. To show Eq. (6.32), consider another arbitrary vector $\mathbf{q} \in \mathbb{R}^m$, $\mathbf{y} = f(\mathbf{x})$ then

$$\frac{d}{dx_i}\langle \mathbf{y}, \mathbf{q}\rangle = \frac{d}{dx_i}\sum_{j=1}^{m} y_j q_j = \sum_{j=1}^{m}\frac{dy_j}{dx_i}q_j, \tag{6.34}$$



(a) Eyes (b) Gaze

(c) Opening mouth (d) Mouth shape

(e) Eyebrows (f) Frown

Figure 6.5: **Interpolation results for image-specific directions.** Direction found using the Jacobian method typically finds highly localized directions specific to a single image such as opening the mouth or eyes, changing the gaze direction, or raising the eyebrows.

which is ith component of $\mathbf{J}^{\mathrm{T}}\mathbf{q}$. Setting $\mathbf{q} = \mathbf{J}\mathbf{v}$ allows us to implement the power iteration algorithm without ever explicitly calculating the Jacobian.

In Paper V, we show that the right-hand singular vectors of $\mathbf{J}_t$ contains semantically meaningful information corresponding to highly localized changes to the generated images. Figure 6.5 shows a selection of edits found using this method.

Classifier Supervision. We further propose to find specific labeled semantic directions using a pretrained attribute classifier as supervision. Concretely we use an attribute classifier released by Lin et al. (2021), which is trained on the 40 binary classes of the CelebA data set (Liu et al., 2015). In addition, we use Hopenet (Ruiz et al., 2018) to predict pose where the network regresses scalar values corresponding

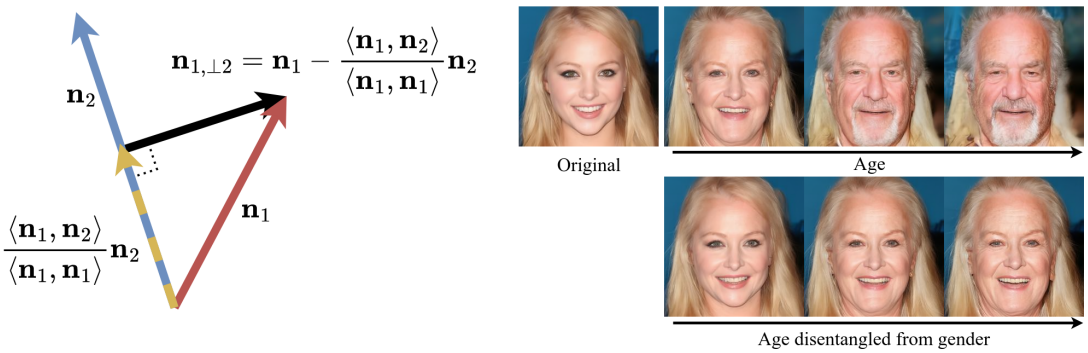


Figure 6.6: **Illustration proposed disentanglement approach.** Given two entangled semantic directions $\mathbf{x}_1$, $\mathbf{x}_2$ we can define a disentangled direction by removing the projection of $\mathbf{x}_2$ onto $\mathbf{x}_1$ from $\mathbf{x}_2$. The right-hand side of the figure illustrates that this approach can effectively remove the entanglement between two found directions corresponding to age and gender.

to yaw, pitch, and roll. Using the attribute networks we annotate random samples from the DDM model. For each sample, we record the bottleneck activation $\mathbf{h}_t$ and define semantic editing directions simply as the difference vector between averages of positive and negative examples from each class. This approach is related to vector arithmetic properties reported for GANs by Radford et al. (2016), but have not previously been shown for DDMs.

We observe that some of our found directions affect several attributes in the image, *e.g.*, a direction for controlling eyeglasses would also make the person in generated image appear older. Inspired by the work of Shen et al. (2020a,b) – which encountered a similar situation for semantic directions found for StyleGAN – we propose to a simple method for disentangling such directions in DDMs. Let $\mathbf{n}_1$ and $\mathbf{n}_2$ denote two semantic directions that are entangled. We can find a new direction $\mathbf{n}_{1,\perp 2}$ that is perpendicular to $\mathbf{n}_2$ by removing the projection of $\mathbf{n}_2$ onto $\mathbf{n}_1$ as

$$\mathbf{n}_{1,\perp 2} = \mathbf{n}_1 - \frac{\langle \mathbf{n}_1, \mathbf{n}_2 \rangle}{\langle \mathbf{n}_1, \mathbf{n}_1 \rangle} \mathbf{n}_1. \tag{6.35}$$

This approach can easily be generalized to remove the effect of several directions as explained in Paper V. Figure 6.6 illustrates this approach and shows the results of disentangling directions corresponding to age and gender.

In summary, Paper V leverages the editing capabilities of the recently proposed semantic latent space (Kwon et al., 2023) in DDMs and proposes several novel methods for discovering interpretable semantic editing directions in this space. To the extent of our knowledge, Paper V provides the first method (concurrently with Park et al. (2023)) for utilizing DDMs for semantic image editing in a fully unsupervised fashion without the need to provide either a semantic mask as a guide as in the work by Couairon et al. (2022) or text-based guidance as in the works by Ramesh et al. (2022), Saharia et al. (2022), and Rombach et al. (2022).

Chapter 7

Conclusions

This final chapter concludes the thesis. The chapter begins in Section 7.1 by addressing some of the numerous ethical and legal issues that arise with the advent of powerful generative models. The chapter ends in Section 7.2 with a discussion of the limitations of the methods proposed in this thesis, perspectives for future work, and a summary of the contributions of this thesis.

7.1 Ethical considerations

The emergence of powerful DGMs such as GANs and DDMs is making it easier to synthesize high-quality images with almost arbitrary content. Additionally, these model enable the editing of existing real images in a way that is very difficult for humans to detect. This raises important ethical issues. In this section, I will outline different categories of ethical issues that arise in the context of modern DGMs capable of image synthesis and editing.

Perpetuating societal biases. The images synthesized by DGMs may reflect and perpetuate biases contained in the training data. In the context of face images, the distribution of images in available large-scale face data sets, such as FFHQ and CelebA (Liu et al., 2015), is biased towards over-representing Caucasian-looking

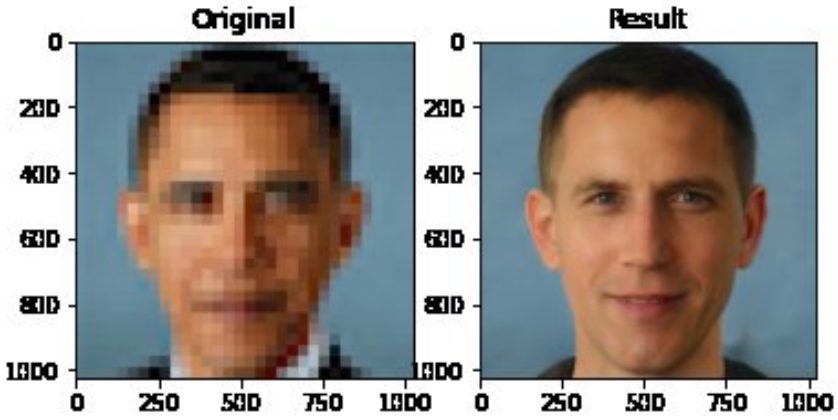


Figure 7.1: **Ethnicity bias in StyleGAN.** The super-resolution system PULSE (Menon et al., 2020) upscales an image of Barack Obama as a Caucasian man. Image from Twitter (Chicken3gg, 2020).

persons. This bias is reflected in the images produced by DGMs trained on these data sets.

An example that gained some attention in the media is the super-resolution system PULSE proposed by Menon et al. (2020). PULSE takes a low-resolution input image and searches the latent space of StyleGAN for a corresponding high-resolution image that, when downsampled, is consistent with the given input image. Several users on Twitter noticed that when inputting low-resolution images of non-Caucasian people, the system tends to produce images of white people Vincent (2020). An example is shown in Fig. 7.1 where the PULSE system upscales a low-resolution image of Barack Obama in a way that results in a Caucasian-looking man.

In connection to the PULSE controversy, Yann LeCun argues on Twitter that the bias is contained in the training data rather than the model:

> ML systems are biased when data is biased. This face upsampling system makes everyone look white because the network was pretrained on FlickFaceHQ, which mainly contains white people pics. Train the *exact* same system on a data set from Senegal, and everyone will look African. (LeCun, 2020)

A related example of bias was found in Shen et al. (2020a,b) where latent directions in StyleGAN found to control age, would also often result in the appearance of eyeglasses in the generated images. This is due to the correlation between age and eyeglasses in the FFHQ training data: older people are more likely to wear glasses. The same bias was found in Paper V in the context of a diffusion model trained on the CelebA data set.

Additional examples in text-to-image diffusion systems were reported in the DALL-E2 model card (Mishkin et al., 2022) where the DALL-E2 system shows strong gender biases when prompted to create images of various professions. For example, when prompting the system to produce images of nurses and flight attendants, DALL-E2 would create images depicting only women in these roles. However, when prompting the system to create images of CEOs and lawyers the generated images would depict only males.

These examples show the importance of recognizing and mitigating the biases that may be present in the training data of image generation systems. As the field advances and these models become more available to the public, it is important and should be a priority of both researchers and companies to ensure that the models adequately reflect the diversity of the real world and do not contribute to perpetuating existing cultural biases.

Secondary and malicious use. The collection of large-scale data sets required for training state-of-the-art models like StyleGAN and Stable Diffusion raises several ethical issues related to privacy, informed consent, and intellectual property rights.

In the case of StyleGAN, the model used in Paper I, Paper II and Paper III is trained on FFHQ which consists of 70k images that was crawled from Flickr. Many of the images were originally uploaded to Flickr by private individuals. Only images under various permissive licenses are used. These licenses allow for the free use, redistribution, and adaptation of the images for non-commercial purposes and in some cases require an indication of which changes have been made and appropriate credit is given to the original author.

Although the images were initially released under these permissive licenses, and as such the users have consented to the use of their images within the scope of the license, the ethical question of informed content is not as clear. With a public platform like Flickr, there is no way of knowing if individuals would have agreed to publish their images if they had known that they would be used for research purposes and for the training of large-scale DGMs.

Although the FFHQ data set is only intended for research purposes, there is a risk that it could be used by malicious actors for unintended and potentially harmful applications. For example, the data set could be exploited to train or enhance facial recognition systems. In the release notice[1] for the FFHQ data set, Nvidia explicitly states that the use of "this data set is not intended for, and should not be used for, development or improvement of facial recognition technologies". However, there is little that prevents malicious actors from misusing the data despite the stated use restrictions.

While the potential for malicious usage of data should be taken into account when releasing large-scale data sets to the public, arguably more severe risks arise from the potential of malicious use of the models themselves. One potential misuse of generative models involves the creation of deceptive deepfakes. Deepfakes are videos or images that have been manipulated using generative models or related technology, possibly with malicious intent. Deepfakes can be used to misrepresent someone as doing or saying something that was not actually done or said thereby spreading misinformation or damaging the reputation of innocent individuals. Deepfakes can range from the relatively benign – as for example the viral images of Pope Francis wearing a Balenciaga-style puffer jacket (Huang, 2023) – to incipiently concerning cases like the viral fake images of Trump getting being arrested (Placido, 2023) to outright dangerous cases as for example the poorly made deepfake that surfaced shortly after the 2022 invasion of Ukraine depicting the Ukrainian President Volodymyr Zelensky asking his soldiers to lay down their weapons (Allyn, 2022).

[1] Available at https://github.com/NVlabs/ffhq-dataset

Attacks on verification systems. Recently, Shmelkin et al. (2021) proposed a generative algorithm for discovering *master faces* using StyleGAN as a face prior. A master face is a face image that passes authentication by an image-based identity authentication system for a large number of stored identities. This shows that the advances in modeling photorealistic human faces might raise security concerns in existing identity verification systems.

Copyright ambiguities. Large-scale text-to-image diffusion systems like Dall-E2, Midjourney, and Stable Diffusion require vast amounts of training data. Recently, the methods by which these data sets have been acquired have come under public scrutiny. As a notable example, consider the current lawsuit from Getty Images against Stability AI, the company behind the release of the Stable Diffusion family of text-to-image diffusion models. In the lawsuit text (Getty Images v. Stability AI, 2023), Stability AI is accused of copyright infringement at a "staggering scale" of the intellectual property right of Getty Images. According to Getty, Stability AI has used more than 12 million images with corresponding captions and metadata collected from Getty without permission.

Another example is the lawsuit by three American artists against Stability AI, Midjourney, and DeviantArt (Butterick, 2023). Systems like Stable Diffusion have the ability to create new images "in the style of" any artists who's work is represented in the training data. For example, it is possible to condition the system to create new images in the style of famous artists like Picasso, Monet, or Edward Munch. However, it is also possible to create new images in the style of contemporary artists. In the lawsuit, it is argued that the training images were used without the consent and without compensating any of those artists thus infringing on their intellectual property rights.

In an interview with the online medium Sifted (Smith, 2023), the CEO of Stability AI stated that the use of the images is protected by "fair use" due to the "transformative" nature of the technology. Here being "transformable", in the context of "fair use" laws, means that the work adds "something new, with a further purpose or different character, and does not substitute for the original use of the work"[2]

[2] https://copyrightalliance.org/faqs/what-is-fair-use/

and therefore changes the nature of the used material.

The central question is thus if these models are truly able to generate novel art or if they are only coping from their training data and, as stated in the court filing, "merely a complex collage tool". It will ultimately be up to governments, international bodies, and the outcomes of court cases like these to decide the future legal framework for the training, use, and distribution these types of models.

Environmental impact. The training of large-scale DGMs requires extensive computational resources. As an example, the reported training compute of StyleGAN was reported to be 41 days and 4 hours (988 hours) on a single Tesla V100 GPU. Using the Machine Learning Impact calculator presented in Lacoste et al. (2019) this amounts to roughly 266 kg of CO2 emitted[3] for training a single high-resolution StyleGAN2 model. For comparison Stable Diffusion 2 was reported to require 200000 GPU hours on a Nvidia A100 GPU, amounting to a total equivalent emission of 15000 kg CO2. For comparison, the average yearly emission per capita in Denmark in 2021 was 5100kg (Ritchie et al., 2020).

7.2 Limitations and future work

A limitation of the tensor model proposed in Paper I and Paper II is that it imposes strong restrictions on which data sets can be used. In particular, the model requires a data set with highly structured labels such that all data is available for each mode in the data tensor. In other words, it is a requirement that for each person, images with all expressions and rotations are available without any missing data. This is a strong requirement that limits the applicability of the method since most data sets do not have such a complete structure.

An avenue for future research could be to formulate the model such that some missing data is allowed using tensor completion methods. This might make the method applicable to less structured data sets. For example, the model could possibly be extended to accommodate the 40 binary attributes from CelebA.

[3] Assuming training on AWS.

In Paper III, we proposed a novel method to utilize NRSfM to get explicit control over the 3D structure of StyleGAN images. However, we only showed results on models trained on human face images. Since the method only requires access to a landmark extractor trained on the domain of the generator, it would be interesting to see if the proposed method can be extended to other structured domains, such as full-body humans or hands.

Further, our approach has only had access to face landmarks as annotation during training. Previously, semantic editing using only landmarks has been described as a challenging application (Wang et al., 2021) since 2D landmark points are extremely localized as compared to more global attributes such as age and gender. Very recently, DragGAN Pan et al. (2023) has made significant progress in point-based semantic editing. Their method allows for fine-grained point-based control of images generated with StyleGAN and allows the user to drag points selected in the image towards new specified target points. It would be interesting to integrate the insights from Pan et al. (2023) into the NRSfM-based approach presented in Paper V.

The discovery that DDMs have a semantic latent space which facilitates semantic editing opens interesting avenues of future research. As mentioned in Paper V, we did preliminary experiments on DDM models trained on churches and bedrooms. However, in their current form, our proposed methods were not able to convincingly discover semantic directions in DDMs that were trained on less structured data sets than human faces. Concurrent work by Park et al. (2023) shows editing results on animal faces in addition to human faces for their unsupervised method. However, the study did not show results on more unstructured domains. Thus, it is still unclear how to develop unsupervised methods for semantic editing in semantic latent space of DDMs when the denoiser is trained on domain that are less structured domains than faces.

Another open question is whether our proposed editing techniques can be transferred to models trained on other modalities than images. For example, it would be interesting to see to which extent our methods can be extended to perform semantically interpretable edits in diffusion models trained to generate video, *e.g.*, those proposed by Ho et al. (2022a,b), or models designed for audio generation.

In summary, this thesis has made several novel contributions to the field of deep generative modeling. Specifically, the thesis has proposed methods aimed at gaining additional semantic control over face images generated with DGMs.

The thesis demonstrates that using a HOSVD-based factorization approach, as described in Paper I and Paper II, can identify semantic directions in the latent space of StyleGAN. These directions control single, semantically meaningful, attributes of generated images. Our multilinear treatment provides a method for factorizing the latent space into different subspaces, each governing different semantic content of the generated images. In particular, the thesis has shown that this approach is capable of finding latent directions that allow for explicit control over facial expressions in generated images.

Furthermore Paper III has contributed to bridging the gap between the explicit 3D control offered by traditional computer graphics techniques and the quality of images generated with data-driven deep generative models such as StyleGAN. Specifically, we have shown that explicit 3D control can be achieved over the generated images by incorporating NRSfM into the image generation process.

Finally, Paper V proposed several novel techniques for discovering meaningful editing directions in the latent space of diffusion models trained on the domain of human face images. In particular, we proposed a method that successfully identifies semantically meaningful edits in an entirely unsupervised fashion. In combination, the methods presented in Paper V can identify editing directions that correspond to a diverse set of semantic changes in the images. These changes include pose, age, gender, and eyeglasses as well as localized changes to specific facial features like the mouth, eyes, eyebrows, and hairstyle.

Bibliography

Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021.

Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18490–18500, 2022. doi: 10.1109/CVPR52688.2022.01796.

Yuval Alaluf, Or Patashnik, Zongze Wu, Asif Zamir, Eli Shechtman, Dani Lischinski, and Daniel Cohen-Or. Third time's the charm? image and video editing with stylegan3. In Leonid Karlinsky, Tomer Michaeli, and Ko Nishino, editors, *Computer Vision – ECCV 2022 Workshops*, pages 204–220, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-25063-7.

Bobby Allyn. Npr: Deepfake video of zelenskyy could be 'tip of the iceberg' in info war, experts warn, 2022. Available online online at: https://www.npr.org/2022/03/16/1087062648/deepfake-video-zelenskyy-experts-war-manipulation-ukraine-russia%7D, last accessed on 28.08.2023.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

Peter Baylies. Stylegan encoder - converts real images to latent space. https://github.com/pbaylies/stylegan-encoder/, 2019.

A.H. Bermano, R. Gal, Y. Alaluf, R. Mokady, Y. Nitzan, O. Tov, O. Patashnik, and D. Cohen-Or. State-of-the-art in the architecture, methods and applications of StyleGAN. *Computer Graphics Forum*, 41(2):591–611, May 2022. doi: 10.1111/cgf.14503.

V Blanz and T Vetter. A morphable model for the synthesis of 3d faces. In *26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1999)*, pages 187–194. ACM Press, 1999.

Sami S Brandt and Hanno Ackermann. Non-rigid structure-from-motion by rank-one basis shapes. *arXiv preprint arXiv:1904.13271*, 2019.

C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, volume 2, page 690–696. IEEE Comput. Soc, 2000. ISBN 978-0-7695-0662-3. doi: 10.1109/CVPR.2000.854941.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, Feb 2019.

Matthew Butterick. Stable diffusion litigation, 2023. Available online online at: https://stablediffusionlitigation.com/pdf/00201/1-1-stable-diffusion-complaint.pdf, last accessed on 17.08.2023.

Twitter User Chicken3gg. Twitter post, 2020. Available online online at: https://twitter.com/Chicken3gg/status/1274314622447820801, last accessed on 07.07.2023.

Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. In *The Eleventh International Conference on Learning Representations*, 2022.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proc. CVPR*, pages 4690–4699, 2019.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021.

Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1532–1545, 2014.

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. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

DC Dowson and BV Landau. The fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, 12(3):450–455, 1982.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022.

Getty Images v. Stability AI. Getty images v. stability ai, 2023. Available online online at: https://fingfx.thomsonreuters.com/gfx/legaldocs/byvrlkmwnve/GETTY%20IMAGES%20AI%20LAWSUIT%20complaint.pdf, last accessed on 17.08.2023.

GH Golub and C Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5):403–420, 1970.

Ian Goodfellow. Twitter post, 2019. Available online online at: https://twitter.com/goodfellow_ian/status/1084973596236144640, last accessed on 07.09.2023.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, page 2672–2680. Curran Associates, Inc., 2014.

Stella Graßhof, Hanno Ackermann, Sami Brandt, and Jörn Ostermann. Apathy is the root of all expressions. *12th IEEE Conference on Automatic Face and Gesture Recognition (FG2017)*, 2017. doi: 10.1109/FG.2017.83.

Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. StyleneRF: A style-based 3d aware generator for high-resolution image synthesis. In *International Conference on Learning Representations*, 2022.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

René Haas, Stella Graßhof, and Sami S. Brandt. Tensor-based subspace factorization for stylegan. In *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, page 1–8. IEEE Press, 2021. doi: 10.1109/FG52635.2021.9666953. URL https://doi.org/10.1109/FG52635.2021.9666953.

René Haas, Stella Graßhof, and Sami S. Brandt. Controllable gan synthesis using non-rigid structure-from-motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 678–687, June 2023a.

René Haas, Stella Graßhof, and Sami Sebastian Brandt. Tensor-based emotion editing in the stylegan latent space. *arXiv:2205.06102 [cs]*, May 2022. URL http://arxiv.org/abs/2111.04554. Accepted for poster presentation at AI4CC @ CVPRW.

René Haas, Inbar Huberman-Spiegelglas, Rotem Mulayoff, and Tomer Michaeli. Discovering interpretable directions in the semantic latent space of diffusion models, 2023b. URL https://arxiv.org/abs/2303.11073.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *Advances in neural information processing systems*, 33:9841–9850, 2020.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022a.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022b.

Kalley Huang. New york times: Why pope francis is the star of a.i.-generated photos, 2023. Available online online at: https://www.nytimes.com/2023/04/08/technology/ai-photos-pope-francis.html, last accessed on 28.08.2023.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *International Conference on Learning Representations*, 2020.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*, pages 4396–4405, 2019. doi: 10.1109/CVPR.2019.00453.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 852–863. Curran Associates, Inc., 2021.

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with

diffusion models. In *Conference on Computer Vision and Pattern Recognition 2023*, 2023.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

Jean Kossaifi. Tensor unfolding, Mar 2017. Available online online at: http://jeankossaifi.com/blog/unfolding.html, last accessed on 20.10.2023.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. In *International Conference on Learning Representations*, 2023.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

Yann LeCun. Twitter post, 2020. Available online online at: https://twitter.com/ylecun/status/1274782757907030016, last accessed on 07.07.2023.

Ji Lin, Richard Zhang, Frieder Ganz, Song Han, and Jun-Yan Zhu. Anycost gans for interactive image synthesis and editing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Yunfan Liu, Qi Li, Qiyao Deng, Zhenan Sun, and Ming-Hsuan Yang. Gan-based facial attribute manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022.

Andrew Melnik, Maksim Miasayedzenkau, Dzianis Makarovets, Dzianis Pirshtuk, Eren Akbulut, Dennis Holzmann, Tarek Renusch, Gustav Reichert, and Helge Ritter. Face generation and editing with stylegan: A survey. *arXiv preprint arXiv:2212.09102*, 2022.

Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 2437–2445, 2020.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

Pamela Mishkin, Lama Ahmad, Miles Brundage, Gretchen Krueger, and Girish Sastry. Dall-e 2 preview - risks and limitations, 2022. Available online online at: https://github.com/openai/dalle-2-preview/blob/main/system-card.md, last accessed on 17.08.2023.

Masahiro Mori, Karl F MacDorman, and Norri Kageki. The uncanny valley [from the field]. *IEEE Robotics & automation magazine*, 19(2):98–100, 2012.

Yael Moses, Shimon Ullman, and Shimon Edelman. Generalization to novel images in upright and inverted faces. *Perception*, 25(4):443–461, 1996.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

Dmitry Nikitko. Stylegan – encoder for official tensorflow implementation. https://github.com/puzer/stylegan-encoder/, 2019.

Yotam Nitzan, Kfir Aberman, Qiurui He, Orly Liba, Michal Yarom, Yossi Gandelsman, Inbar Mosseri, Yael Pritch, and Daniel Cohen-Or. Mystyle: A personalized generative prior. *arXiv preprint arXiv:2203.17272*, 2022.

Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. Drag your gan: Interactive point-based manipulation on the generative image manifold. *arXiv preprint arXiv:2305.10973*, 2023.

Yong-Hyun Park, Mingi Kwon, Junghyo Jo, and Youngjung Uh. Unsupervised discovery of semantic latent directions in diffusion models. *arXiv preprint arXiv:2302.12469*, 2023.

Roger Penrose. *A generalized inverse for matrices.* Mathematical Proceedings of the Cambridge Philosophical Society, 1955. ISBN 1600490069, 9781600490064. doi: 10.1017/S0305004100030401.

Dani Di Placido. Forbes: Ai-generated images of donald trump getting arrested foreshadow a flood of memes, fake news, 2023. Available online online at: https://www.forbes.com/sites/danidiplacido/2023/03/22/ai-generated-images-of-donald-trump-getting-arrested-foreshadow-a-flood-of-memes-fake-news, last accessed on 28.08.2023.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and

et al. Learning transferable visual models from natural language supervision. In *Proc. ICML*, Feb 2021. arXiv: 2103.00020.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022. doi: 10.48550/arXiv.2204.06125.

Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: A stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2287–2296, June 2021.

Hannah Ritchie, Max Roser, and Pablo Rosado. Co2 and greenhouse gas emissions. *Our World in Data*, 2020. https://ourworldindata.org/co2-and-greenhouse-gas-emissions.

Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Trans. Graph.*, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

Nataniel Ruiz, Eunji Chong, and James M. Rehg. Fine-grained head pose estimation without keypoints. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on*

Computer Vision and Pattern Recognition (CVPR), pages 22500–22510, June 2023.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, 2022.

Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1532–1540, June 2021.

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020a.

Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *TPAMI*, 2020b.

Ron Shmelkin, Tomer Friedlander, and Lior Wolf. Generating master faces for dictionary attacks with a network-assisted latent space evolution. In *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, pages 01–08. IEEE, 2021.

Edgar Simo-Serra, Satoshi Iizuka, Kazuma Sasaki, and Hiroshi Ishikawa. Learning to simplify: fully convolutional networks for rough sketch cleanup. *ACM Transactions on Graphics (TOG)*, 35(4):1–11, 2016.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

L Sirovich and M Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America. A, Optics and image science*, 4(3):519–524, 1987.

Christian Sivertsen, René Haas, Halfdan Hauch Jensen, and Anders Sundnes Løvlie. Exploring a digital art collection through drawing interactions with a deep generative model. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI EA '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394222. doi: 10.1145/3544549.3583902. URL https://doi.org/10.1145/3544549.3583902.

Tim Smith. Openai's actions 'fundamentally wrong', says stability ai's emad mostaque, 2023. Available online online at: https://sifted.eu/articles/stable-diffusion-ai-emad-mostaque, last accessed on 17.08.2023.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zöllhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images, cvpr 2020. In *Proc. CVPR)*. IEEE, june 2020.

Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992. doi: 10.1007/BF00129684.

Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Trans. Graph.*, 40(4), jul 2021. ISSN 0730-0301. doi: 10.1145/3450626.3459838.

Emanuele Trucco and Alessandro Verri. *Introductory techniques for 3-D computer vision*, volume 201. Prentice Hall Englewood Cliffs, 1998.

Raffaele Tucciarelli, Neza Vehar, Shamil Chandaria, and Manos Tsakiris. On the realness of people who do not exist: The social processing of artificial faces. *Iscience*, 25(12):105441, 2022.

Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

M. A. O. Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, ECCV '02, page 447–460, Berlin, Heidelberg, 2002. Springer-Verlag. ISBN 3540437452.

M Alex O Vasilescu and Demetri Terzopoulos. Multilinear subspace analysis of image ensembles. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–93. IEEE, 2003.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

James Vincent. The verge: What a machine learning tool that turns obama white can (and can't) tell us about ai bias, 2020. Available online online at: https://www.theverge.com/21298762/face-depixelizer-ai-machine-learning-tool-pulse-stylegan-obama-bias, last accessed on 07.09.2023.

Hui-Po Wang, Ning Yu, and Mario Fritz. Hijack-gan: Unintended-use of pretrained, black-box gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7872–7881, June 2021.

Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12863–12872, June 2021.

Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and M.J. Rosato. A 3d facial expression database for facial behavior research. In *7th Intern. Conf. on Automatic Face and Gesture Recognition (FGR06)*, pages 211–216, 2006. doi: 10.1109/FGR.2006.6.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Jiapeng Zhu, Yujun Shen, Yinghao Xu, Deli Zhao, and Qifeng Chen. Region-based semantic factorization in GANs. In *International Conference on Machine Learning (ICML)*, 2022.

Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 597–613. Springer, 2016.

Appendix A

Papers

A.1 Paper I: Tensor-based Subspace Factorization for StyleGAN

Tensor-based Subspace Factorization for StyleGAN

René Haas, Stella Graßhof and Sami S. Brandt
IT University of Copenhagen, Copenhagen, Denmark

***Abstract*— In this paper, we propose τGAN a tensor-based method for modeling the latent space of generative models. The objective is to identify semantic directions in latent space. To this end, we propose to fit a multilinear tensor model on a structured facial expression database, which is initially embedded into latent space. We validate our approach on StyleGAN trained on FFHQ using BU-3DFE as a structured facial expression database. We show how the parameters of the multilinear tensor model can be approximated by Alternating Least Squares. Further, we introduce a stacked style-separated tensor model, defined as an ensemble of style-specific models to integrate our approach with the extended latent space of StyleGAN. We show that taking the individual styles of the extended latent space into account leads to higher model flexibility and lower reconstruction error. Finally, we do several experiments comparing our approach to former work on both GANs and multilinear models. Concretely, we analyze the expression subspace and find that the expression trajectories meet at an apathetic face that is consistent with earlier work. We also show that by changing the pose of a person, the generated image from our approach is closer to the ground truth than results from two competing approaches.**

I. Introduction

In this paper, we propose a novel framework for finding semantic directions in the latent space of Generative Adversarial Networks (GANs) [10]. GANs have, since their proposal, emerged as one of the most dominant approaches for unsupervised representation learning in Computer Vision and beyond [23].

Architecturally GANs refer to the simultaneous training of two neural networks: a *generator* and a *discriminator*. The generator produces images by sampling from its latent space, while the discriminator, a binary classifier, tries to discriminate the generated images from the training images. The goal of training is to reach the equilibrium of the min-max game between the two adversaries, such that neither can improve by changing the parameter values. At equilibrium, the discriminator can be discarded, and the generator can then be used to produce new data by sampling from the latent distribution. The new data points follow the same statistics as the training data but are not contained in it. Modern state-of-the-art GAN variations have borrowed from the Style-transfer literature [14], [22] to disentangle the latent space and synthesize high-quality face images. Work by [17], [18], and most recently [16], showed how to a train state-of-the-art StyleGAN model, even in cases of limited data.

A recent goal has been to find semantically interpretable directions in GAN latent spaces, and several approaches for *semantic face editing* have been proposed. Semantic face editing refers to the ability to change various semantic attributes, such as identity, expression, and rotation, gender, of the generated images. Early work used an information criterion (InfoGAN) [6] to determine semantic directions. However, as pointed out in [8], there is no guarantee that the latent codes produced by this method are semantically meaningful. Additional unsupervised approaches for finding semantic directions in StyleGAN include Principal Component Analysis (PCA) on sampled latent codes [15] and the closed-form factorization suggested by [25].

978-1-6654-3176-7/21/$31.00 ©2021 IEEE

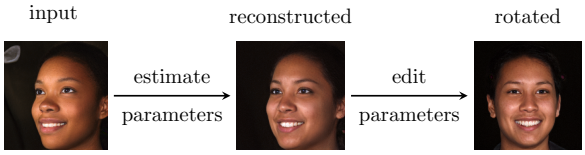


Fig. 1: Overview of the proposed approach.

A recent approach for finding semantic directions in StyleGAN in a supervised fashion is to train binary linear classifiers (SVMs) to detect single binary semantic attributes such as smile vs. no smile, male vs. female, glasses vs. no glasses. For a given semantic attribute, the semantic direction could then be defined as the normal to the supporting hyperplanes of the trained SVM [24].

In the literature, a wide collection of *multilinear* methods have been proposed to model and analyze faces and expressions. Early, PCA or dictionary-based 3D Morphable Models (3DMM) [3], [9] capture the variation in shape and texture of neutral 3D faces. Recently 3DMMs have also been used to make semantic edits to images generated by StyleGAN [27]. More recently, factorization methods, based on higher-order data representations, were introduced with the benefit of better disentanglement of dimensions, such as person-specific shape and expression, when compared to matrix methods [28], [30]. These models were built on the Higher-Order Singular Value Decomposition (HOSVD) to factorize the data, and have successfully been used to model faces, their 3D reconstruction, as well as in transferring expressions [4], [5]. Moreover, in [11], [12] a HOSVD tensor model was constructed from the Binghamton 3D facial expression database (BU-3DFE) [33], which revealed a practically planar expression subspace, in which the six basic emotions form one-dimensional affine subspaces [11]. These six lines intersect in a common vertex, the origin of expressions, which surprisingly does not represent the neutral face, but an extrapolated expression referred to as *apathetic*.

The main novelty of this work is to use a multilinear face model to analyze the latent space of GANs. More specifically, we propose to use the HOSVD to factorize the

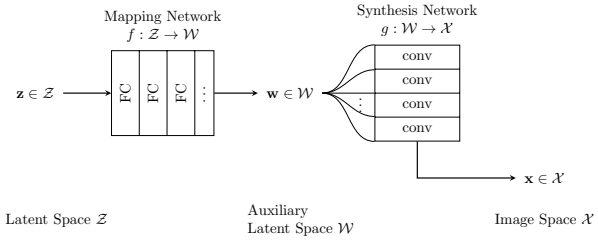


Fig. 2: Architecture of the StyleGAN generator.

latent space into semantically meaningful linear subspaces that yield a multilinear tensor model. Given an input image, we estimate the model parameters to approximate the input, and then change one attribute, such as rotation, as illustrated in Fig. 1.

The main contributions of this paper are as follows:

- We propose a novel method for semantic face editing with StyleGAN.
- We propose a method to estimate model parameters and present reasonable regularization, enabling stable parameter transfer.
- We show that expression trajectories intersect at a unique point, corresponding to the origin of expressions, which differs from the neutral face confirming the earlier findings [11], [12] based on BU-3DFE.
- We propose an extended model, based on style separation, which leads to greater model flexibility and lower reconstruction error for independent test images.

The paper is organized as follows: In Sec. II we will review the architecture and outline the process on how to embed reference images into the latent space of StyleGAN. In Sec. III we present our Tensor-Based GAN model which we build "on top of" the StyleGAN latent space. Here we will also elaborate on how we can approximate model parameters for a given latent vector. Experiments and results of our proposed approach are presented in Sec. IV, followed by a summary and conclusion in Sec. V.

II. StyleGAN

In this section, we will review the StyleGAN architecture and explain how to embed reference images into the latent space of the pre-trained models released by Nvidia [17], [18].

A. StyleGAN Architecture

The StyleGAN *generator* $G : \mathcal{Z} \to \mathcal{X}$, where $G = g \circ f$, is composed of two networks, the *mapping network* $f : \mathcal{Z} \to \mathcal{W}$ and the *synthesis network* $g : \mathcal{W} \to \mathcal{X}$, see Fig. 2. The mapping network f, maps the latent vector $\mathbf{z} \in \mathcal{Z}$ onto the auxiliary latent space $\mathcal{W}$ to the vector $\mathbf{w} = f(\mathbf{z})$ while the synthesis network $g : \mathcal{W} \to \mathcal{X}$ maps the vector $\mathbf{w} \in \mathcal{W}$ to the final output image $\mathbf{x} \in \mathcal{X}$ in image space. The full generator G thus maps the latent vector $\mathbf{z}$ to an image $\mathbf{x}$. The notation used in this paper is summarized in Tab. I.

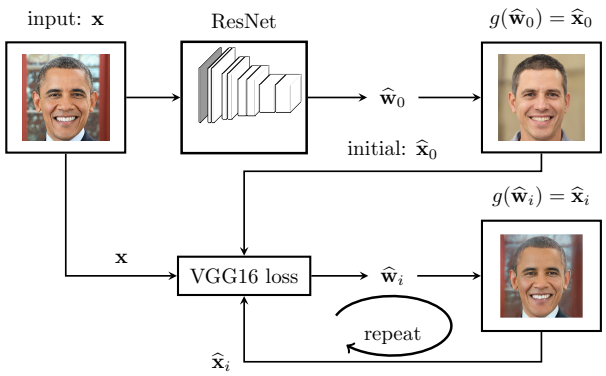


Fig. 3: Diagram illustrating image embedding into the auxiliary latent space $\mathcal{W}$.

B. Generator Inversion

GANs do not include an encoder as part of their architecture. Therefore, a goal in GAN research has been to find a method for finding a latent code that produces an image as close as possible to a given reference image, which we refer to as *embedding* an image into the latent space. The problem can be considered as inverting the synthesis network $g^{-1} : \mathcal{X} \to \mathcal{W}$ [1], [21] while inverting G, and thereby embedding into $\mathcal{Z}$ space, has been investigated in [18]. Contemporary techniques for $\mathcal{W}$ space embedding, i.e. finding g^{-1}, use a VGG network [26]. Our approach for embedding onto the auxiliary latent space $\mathcal{W}$ is illustrated in Fig. 3. The inverse generator $G^{-1} : \mathcal{X} \to \mathcal{Z}$ yields the latent vector $\mathbf{z} = G^{-1}(\mathbf{x})$ with $G^{-1} = f^{-1} \circ g^{-1}$ for the input image $\mathbf{x}$.

The initial estimate for the auxiliary latent vector for a given reference image is computed as follows. We use the pre-trained weights of StyleGAN [17] and the recently revised StyleGAN2 [18] architecture. Then, as proposed in [2], we train a ResNet [13] in a supervised setting using synthetic StyleGAN data to approximate g^{-1} that yields the initial estimate $\widehat{\mathbf{w}}_0$ for the latent vector. The refinement for the auxiliary latent vector is computed by first using the VGG16 network [26], pre-trained on ImageNet database, and then removing the classification layer, hence the truncated network produces a high dimensional feature vector for a given input image, as described in [34]. Since the trained generator is fully differentiable, the loss can be calculated in VGG space and gradients back-propagated through the generator, hence we can iteratively update the latent code. This approach is also used in [21]. We also found that using the ResNet estimate as initialization for the VGG optimization process, leads to faster convergence than not using ResNet initialization.

III. Multilinear Model

This section introduces τGAN, our latent space factorization method for GANs that augments the StyleGAN synthesis network g with a multilinear tensor model. We do this by embedding a facial expression database into the

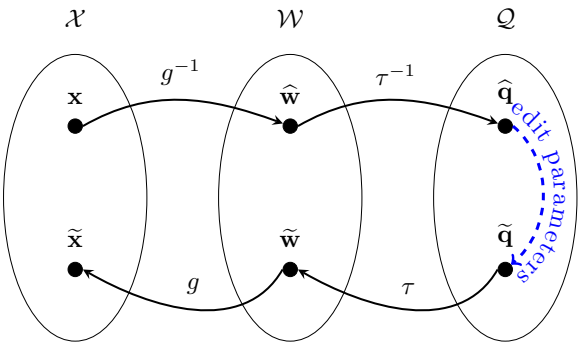


Fig. 4: Overview of the different spaces and how the function relate them, c.f. Tab. I. The blue line indicates a manual change of one of the parameter vectors for transfer of person, expression or rotation.

TABLE I: Overview of the notation used in this work.

Symbol	Description
$\mathcal{X}$	Image Space
$\mathcal{Z}$	Latent Space
$\mathcal{W}$	Auxiliary Latent Space
$\mathcal{Q}$	Parameter Space
Operator	**Name**
$f : \mathcal{Z} \to \mathcal{W}$	Mapping Network
$g : \mathcal{W} \to \mathcal{X}$	Synthesis Network
$g^{-1} : \mathcal{X} \to \mathcal{W}$	StyleGAN Embedder
$\tau^{-1} : \mathcal{W} \to \mathcal{Q}$	Parameter Estimator
$\tau : \mathcal{Q} \to \mathcal{W}$	Tensor Model

auxiliary latent space $\mathcal{W}$ of StyleGAN. We then order the embedded database into a tensor, which we factorize into semantic subspaces. The resulting parameter space $\mathcal{Q}$ will thus be the Cartesian product of the semantic subspaces $\mathcal{Q} = \mathcal{Q}_\mathrm{P} \times \mathcal{Q}_\mathrm{E} \times \mathcal{Q}_\mathrm{R}$, where $\mathcal{Q}_\mathrm{P}$ is the person space, $\mathcal{Q}_\mathrm{E}$ the expression space, and $\mathcal{Q}_\mathrm{R}$ is the rotation subspace. An overview of the different spaces and how the operators relate them are displayed in Fig. 4 and Tab I.

A. Tensor Factorization

The Higher-Order Singular Value Decomposition (HOSVD) is a generalization of the matrix SVD to higher-order tensors [7], [32], [11], [29], [28], [19].

The starting point for our analysis is a standardized data tensor $T \in \mathbb{R}^{N \times P \times E \times R}$, where N refers to the number of elements in the latent vector, P is the number of people, E the number of expressions, and R number of viewpoints or rotations. Using the HOSVD T can then be factorized as

$$T \simeq C \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3 \times_4 \mathbf{U}_4, \tag{1}$$

where $\times_k$ denotes the k-way product, $C \in \mathbb{R}^{\tilde{N} \times \tilde{P} \times \tilde{E} \times \tilde{R}}$ is the core tensor, and $\mathbf{U}_1 \in \mathbb{R}^{N \times \tilde{N}}$, $\mathbf{U}_2 \in \mathbb{R}^{P \times \tilde{P}}$, $\mathbf{U}_3 \in \mathbb{R}^{E \times \tilde{E}}$, $\mathbf{U}_4 \in \mathbb{R}^{R \times \tilde{R}}$ are matrices with orthonormal columns constructed from the singular vectors of the k-mode matrix unfoldings of T. In general we have that $\tilde{N} \leq N$, $\tilde{P} \leq P$, $\tilde{E} \leq E$, and $\tilde{R} \leq R$.

B. Multilinear Tensor Model for GANs

The HOSVD (1) factorizes the data tensor into a core tensor, and a set of factor matrices $\mathbf{U}_i$, one for each subspace. By selecting appropriate *rows* from $\mathbf{U}_i$, $i = 2, 3, 4$, one normalized latent vector, i.e. a single mode-1 fiber of T, can be recovered. For example, to recover the latent vector of person p performing expression e with rotation r, the p^{th} row of $\mathbf{U}_2$, e^{th} row of $\mathbf{U}_3$, and r^{th} row of $\mathbf{U}_4$ is selected. This can be conveniently formulated by a *canonical basis*, where the parameter vectors $\mathbf{q}'_2 \in \mathbb{R}^P$, $\mathbf{q}'_3 \in \mathbb{R}^E$ and $\mathbf{q}'_4 \in \mathbb{R}^R$ pick a weighted linear combination of the rows of the $\mathbf{U}_i$ matrices. Therefore, a given latent code $\mathbf{y}'$ can be approximated by the model prediction $\widehat{\mathbf{y}}'$ as

$$\widehat{\mathbf{y}}' = C \times_1 \mathbf{U}_1 \times_2 \mathbf{q}_2'^{\mathrm{T}} \mathbf{U}_2 \times_3 \mathbf{q}_3'^{\mathrm{T}} \mathbf{U}_3 \times_4 \mathbf{q}_4'^{\mathrm{T}} \mathbf{U}_4. \tag{2}$$

This expression can be further simplified by defining $\mathbf{q}_i^{\mathrm{T}} \equiv \mathbf{q}_i'^{\mathrm{T}} \mathbf{U}_i$ and analogously $\widehat{\mathbf{y}} = \mathbf{U}_1^{\mathrm{T}} \widehat{\mathbf{y}}'$. Now applying $\times_1 \mathbf{U}_1^{\mathrm{T}}$ to both sides of (2) and recalling that the columns the respective $\mathbf{U}$ matrices are orthonormal we can write a more compact model representation as

$$\widehat{\mathbf{y}} = C \times_2 \mathbf{q}_2^{\mathrm{T}} \times_3 \mathbf{q}_3^{\mathrm{T}} \times_4 \mathbf{q}_4^{\mathrm{T}}, \tag{3}$$

where the unprimed coordinates refer to the latent code in the eigenspace spanned by the columns of the $\mathbf{U}_i$ matrices. In this formulation, we have 3 individual parameter vectors and use repeated n-mode products to relate these to the model prediction.

We can rewrite (3) in a more general form to illustrate the mathematical structure of our model. Let us define the $P \times E \times R$, rank-1 parameter tensor $Q = \mathbf{q}_2^{\mathrm{T}} \otimes \mathbf{q}_3^{\mathrm{T}} \otimes \mathbf{q}_4^{\mathrm{T}}$, where $\otimes$ refers to the tensor product. Then the components of the rank-1 parameter tensor $Q \in \mathbb{R}^{P \times E \times R}$ is given by $Q_{\nu\rho\lambda} = q_\nu^{(2)} q_\rho^{(3)} q_\lambda^{(4)}$ where $q_\nu^{(k)}$ refers to the νth component of the subspace vector $\mathbf{q}_k \in \mathcal{Q}_k$ for $k = \{2, 3, 4\}$.

With this definition, we can write (3) in a more compact and convenient representation using the Einstein summation convention

$$\hat{Y}^\mu = C^{\mu\nu\rho\lambda} Q_{\nu\rho\lambda}. \tag{4}$$

This lets us write the latent code, in the auxiliary latent space $\mathcal{W}$, as an application of the multilinear map, defined by the core tensor C, on the parameter tensor Q.

Our entire tensor model τ can thus be written as the composite map of the core C followed by the change-of-basis transformation defined by $\mathbf{U}_1 : \mathcal{W} \to \mathcal{W}$, and the inverse standardization operator $\Omega^{-1} : \mathcal{W} \to \mathcal{W}$, where Ω^{-1} translates and scales a latent vector back to the original scale of $\mathcal{W}$ space according to the mean and variance of the BU-3DFE data.

C. Stacked Style-Separated Model

In addition to the previously presented model, we propose an alternative approach, where styles are separated instead of vectorizing the latent code. That is, we interpret the S styles of $\mathbf{w}$ as separate vectors of dimension L, which is also indicated in Fig. 2. To separate the S styles, we

propose to order the latent codes into the data tensor $T_{\text{style}} \in \mathbb{R}^{S\times L\times P\times E\times R}$.

Then the shape dimension can be addressed separately by defining the style-specific tensors

$$T_s \in \mathbb{R}^{L\times P\times E\times R}, \quad s = 1, 2, \cdots, S. \tag{5}$$

We factorize each style-specific tensor T_s, and define style-specific tensor model τ_s. The ensemble of these models is referred to as the *stacked style-separated model* τ_S, which has $S(P+E+R)$ parameters. In conclusion, while the prior vectorized model τ, based on T, has $P+E+R$ parameters, this formulation τ_S has $S(P+E+R)$ parameters since it models the style separately.

D. Optimization

Our next aim is to estimate the model parameters by constructing the estimator $\tau^{-1} : \mathcal{W} \to \mathcal{Q}$. The estimator is defined as the solution to the optimization problem

$$\begin{aligned} &\min_{Q} ||\widehat{\mathbf{y}} - \mathbf{y}||_2^2 \quad \text{subject to} \\ &||\mathbf{q}_i||_2^2 \leq c_2 \quad \text{and} \\ &||\mathbf{U}_i\mathbf{q}_i||_1 \leq c_1 \quad \text{for} \quad i = 2, 3, 4. \end{aligned} \tag{6}$$

The form of (6) is inspired by [11], [12], and enforces constraints on the model parameters to retrieve a stable representation of new latent vectors by linear combinations within the training data. We regularize the model using Ridge and Lasso regression. Then the Lagrangian for the constrained problem (6) can be written as

$$\begin{aligned} \mathcal{L}(Q, \lambda_1, \lambda_2) = &||\widehat{\mathbf{y}} - \mathbf{y}||_2^2 \\ &+ \sum_{k=2}^{4} \lambda_{2,k} ||\mathbf{q}_k||_2^2 + \lambda_{1,k} ||\mathbf{q}'_k||_1 \end{aligned} \tag{7}$$

where $\lambda_{1,k}, \lambda_{2,k} \geq 0$ refer to regularization parameters, i.e. Lasso and Ridge. Note that there is no prime on the Ridge term since $||\mathbf{q}'_i||_2^2 = (\mathbf{U}_i^{\mathrm{T}}\mathbf{q}_i)^{\mathrm{T}}(\mathbf{U}_i^{\mathrm{T}}\mathbf{q}_i) = ||\mathbf{q}_i||_2^2$ since $\mathbf{U}_i^{\mathrm{T}}\mathbf{U}_i = \mathbf{I}$. We will now continue to present a strategy for solving the constrained optimization problem in (6) by Alternating Least Squares.

As in [11], [12] the minimization can be solved by first rewriting (3) as a matrix-vector multiplication separately for each of the three model parameter vectors as

$$\hat{\mathbf{y}} = \mathbf{A}^{(k)}\mathbf{q}_k, \ k = 2, 3, 4, \tag{8}$$

where the matrices $\mathbf{A}^{(k)}$ are given by

$$\mathbf{A}^{(2)} = C \times_3 \mathbf{q}_3^{\mathrm{T}} \times_4 \mathbf{q}_4^{\mathrm{T}}, \tag{9}$$

$$\mathbf{A}^{(3)} = C \times_2 \mathbf{q}_2^{\mathrm{T}} \times_4 \mathbf{q}_4^{\mathrm{T}}, \tag{10}$$

$$\mathbf{A}^{(4)} = C \times_2 \mathbf{q}_2^{\mathrm{T}} \times_3 \mathbf{q}_3^{\mathrm{T}}. \tag{11}$$

Therefore, an unknown latent vector $\mathbf{y}$ can be estimated by alternating between the systems (8), while updating the matrices $\mathbf{A}^{(k)}$ in each step.

IV. Experiments

In the following, we give some additional details for the BU-3DFE database and continue to report on our experimental results.

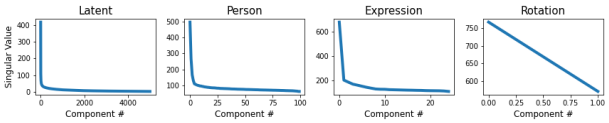


Fig. 5: Validation of decomposition results. Energy of singular values for each mode of T.

A. Facial Expression Database

As mentioned in the introduction, we use the BU-3DFE database [33]. The database contains 3D face scans and images of 100 persons (56 female and 44 male), with varying ages (18-70 years) and diverse ethnic/racial ancestries. Each subject was asked to perform the six basic emotions: anger, disgust, happiness, fear, sadness, and surprise, each with four levels of intensity. Additionally, for each participant, the neutral face was recorded. Hence, for each person, there are a total of 25 facial expressions recorded from two pose directions, left and right, resulting in 5000 face images.

B. Data Prepossessing

As a pre-processing step, we embedded each face image from the BU-3DFE database, into the latent space of StyleGAN, as described in Sec. II-B. We then collected the resulting latent vectors into the 4-way data tensor $T_0 \in \mathbb{R}^{N\times P\times E\times R}$. We then calculated the mode-1 unfolding $\mathbf{T}_0^{(1)} \in \mathbb{R}^{N\times PER}$ of T_0 containing all the PER latent vectors. We then standardized this matrix to zero mean and unit variance for each latent variable and then finally folded this standardized matrix into a $N \times P \times E \times R$ dimensional tensor T which we used for all subsequent experiments.

C. Subspace Analysis

The standardized tensor T was factorized by the HOSVD, as described in (1), yielding the four subspaces spanned by the columns of $\mathbf{U}_k$, $k = 1, \ldots, 4$. The distribution of the energy of the subspaces is shown in Fig. 5, which illustrates the compactness of the subspaces.

In Fig. 6 we show a visualization of the expression subspace. As an initial step, we truncated the expression subspace from 25 dimensions to 3D. It can be seen that for each emotion, the variation in expression strength forms linear trajectories in expression space. These trajectories are star-shaped and meet at an origin of expression which is shared by all emotion trajectories. This is neither the neutral nor the mean face, but the "apathetic" face, found in [11], [12], see Fig. 7(a)-(c). In this case, the apathetic face in Fig. 7(c) is closer to the mean face than in [11], [12], displayed in Fig. 7(f) for comparison.

D. Vectorized vs. Stacked Style-Separated Model

In Sec. III we proposed to build two different versions of tensor models. (1) The *vectorized model* flattens each latent code of one image and then orders them into the tensor $T \in \mathbb{R}^{N\times P\times E\times R}$, and (2) the *stacked style-separated model* $T_{\text{style}} \in \mathbb{R}^{S\times L\times P\times E\times R}$ which considers the $S = 18$ styles of StyleGAN separately. We estimated the parameters for the

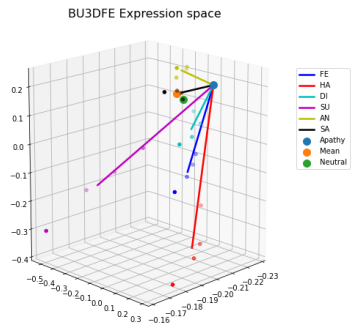
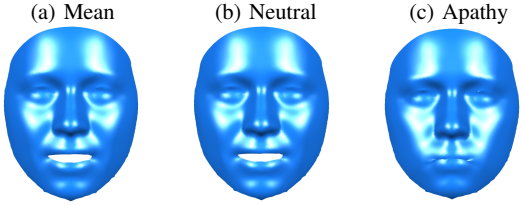


Fig. 6: Projection of the expression subspace, defined by $\mathbf{U}_3$, onto 3 dimensions.

(a) Mean (b) Neutral (c) Apathy



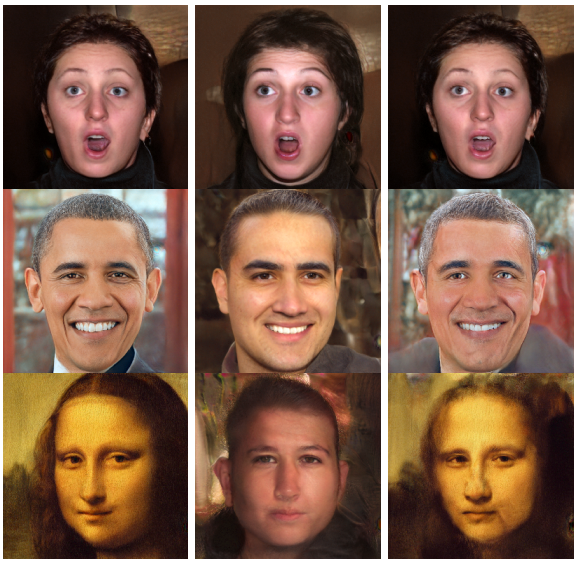
(d) Mean (e) Neutral (f) Apathy

Fig. 7: Synthesized faces for (a) the mean face, (b) neutral face, and (c) apathetic face. Accordingly, (d), (e), (f) show the 3D faces synthesized by the method in [11].

two models, using the ALS procedure (8). The results are illustrated in Fig. 8. It can be seen, that the ground truth (Fig. 8a), is visually closer to the stacked style-separated model (Fig. 8c) than the vectorized model (Fig. 8b) for test images from the BU-3DFE data set (top row), as well as for arbitrary images (2nd and 3rd row). We conclude that the proposed adaptation by the separate styles improves performance.

E. Validation of Regularization Parameters

The optimization problem defined in (7) contains six regularization parameters $\lambda_{1,k}$ and $\lambda_{2,k}$, $k = 2, 3, 4$, two for each of the three parameter vectors, which must be manually set. In the following experiment we investigated how the hyperparameters influenced the quality of the results, and assume that they are the same for the three parameters, hence $\lambda_1 = \lambda_{1,k}$, and $\lambda_2 = \lambda_{2,k}$. Here we used the vectorized model on the basis of the standardized latent codes in (3). Initially, we divided the data into a training, validation, and test set by a randomized 90–5–5 split over the $P = 100$ person identities. The validation set thus had a total of $5ER = 250$ samples. We estimated the tensor model based



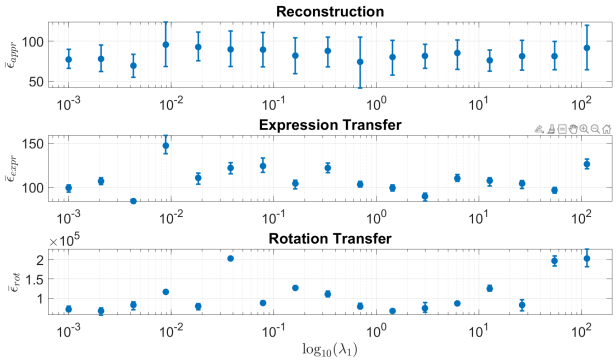
(a) Reference (b) Vectorized (c) Style-Separated

Fig. 8: Reconstructions: (a) Ground truth images, and the results from either (b) the vectorized model, and (c) the style-separated model. The top row shows an example from the BU-3DFE database, while the 2nd and 3rd rows illustrate reconstruction of novel images which are not part of BU-3DFE.

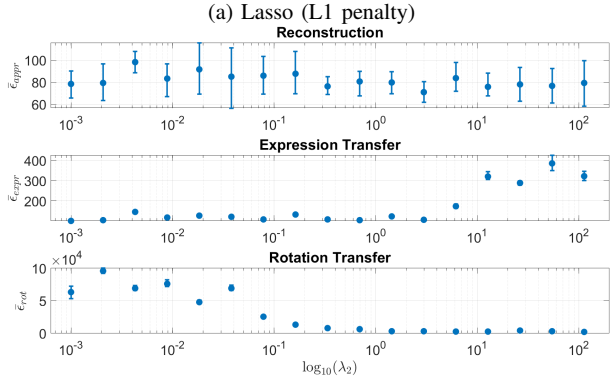
on the training set. For each latent vector in the validation set we then estimated the subspace parameters $\mathbf{q}_i$ by ALS using (8).

We evaluated three kinds of errors for the validation set: the approximation error, and the expression and rotation transfer errors. The approximation error between the ground truth $\mathbf{y}$ and estimated latent code $\widehat{\mathbf{y}}_i$ is defined as $\epsilon_{\text{approx}} = ||\widehat{\mathbf{y}} - \mathbf{y}||_2^2$. The transfer errors result from exchanging estimated parameters $\widehat{\mathbf{q}}_k$ by known values $\widetilde{\mathbf{q}}_k$. Hence using $\widetilde{\mathbf{y}}_{\text{expr}} \equiv \tau(\widehat{\mathbf{q}}_{\text{person}} \otimes \widetilde{\mathbf{q}}_{\text{expr}} \otimes \widehat{\mathbf{q}}_{\text{rot}})$ gives rise to an expression transfer error which we define as $\epsilon_{\text{expr}} = ||\widetilde{\mathbf{y}}_{\text{expr}} - \mathbf{y}||_2^2$. Analogously, the rotation transfer error is defined as the error arising from only changing the parameters associated with the rotation subspace according to $\epsilon_{\text{rot}} = ||\widetilde{\mathbf{y}}_{\text{rot}} - \mathbf{y}||_2^2$. The three error metrics ϵ_{approx}, ϵ_{expr}, and ϵ_{rot} were then calculated for each sample, with varying hyperparameter values λ_1 and λ_2. In this experiment, we investigate Lasso and Ridge regression independently, i.e., we set $\lambda_1 = 0$ while varying λ_2, and vice versa. We restrict ourselves to only consider cases where the regularization strength is equal for all subspaces.

The results are illustrated in Fig. 9. In general, it can be seen that the approximation error is more stable than the other two errors. Fig. 9a suggests that high values of λ_1 should be chosen for rotation transfer, while for expression transfer $\lambda_1 \approx 1$ seems to be a reasonable choice. Fig. 9b reveals that for $\lambda_2 \approx 1$ all error metrics are small, and hence this interval is a good choice.



(a) Lasso (L1 penalty)



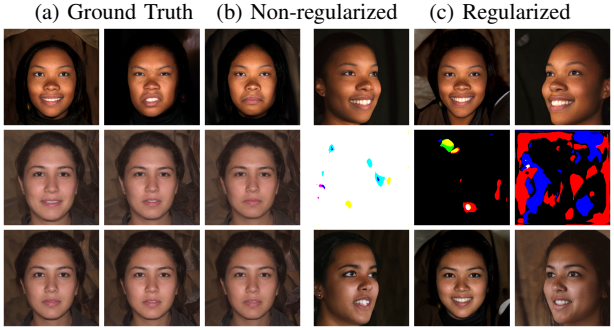
(b) Ridge (L2 penalty)

Fig. 9: Influence of the hyper parameters, λ_1 and λ_2 steering the (a) Lasso and (b) Ridge constraints, on (from top to bottom row) the approximation error, expression transfer error, and rotation transfer error.

F. Regularization and Parameter Transfer

We used the regularization parameters above to perform expression and rotation transfer on samples from the test set. We then synthesized images from the estimated parameters by applying the composite transformation $\widehat{\mathbf{x}} = g(\tau(\widehat{Q}))$ to the estimated subspace parameters $\widehat{Q}$. Additionally, we performed expression and rotation transfer by replacing one of the three estimated parameter vectors by known values, as described before. We did this for the regularized model ($\lambda_1 > 0,\ \lambda_2 > 0$) and the non-regularized model ($\lambda_1 = \lambda_2 = 0$). Fig. 10 shows how well the ground truth, in $\mathcal{W}$ space, (Fig. 10a) can be approximated by the non-regularized solution (Fig. 10b) and the regularized solution (Fig. 10c). It seems that the non-regularized solution matched the ground truth slightly better with respect approximation expression transfer. However, for rotation transfer (Fig. 10e) the regularized solution clearly outperformed the non-regularized solution. Because in the non-regularized solution the resulting image is not recognizable as a face anymore at all, while the regularized solution is not deformed and the rotation of the depicted faces conform to ground truth. This experiment thus showed that adding a small L2 regularization term yields stable rotation transfer.

(a) Ground Truth (b) Non-regularized (c) Regularized



(d) Expression Transfer (e) Rotation Transfer

Fig. 10: Reconstruction and regularization results. (a) Ground truth (b) approximation by the non-regularized model, and (c) the regularized model. (d,e) Results from rotation and expression transfer containing ground truth (top row), the non-regularized solutions (middle row), and the regularized solution (bottom row).

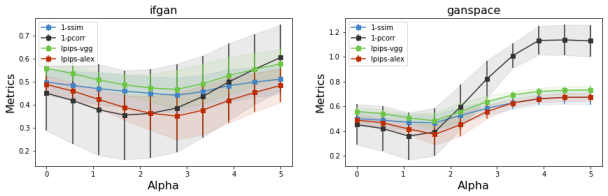


Fig. 11: To find the optimal interpolation strength α for rotation transfer for InterFaceGAN [24] and GANSpace [15] we compare the images generated by shifting the latent code corresponding to an image from the one rotation towards the other and compare the result with the ground truth.

G. Quantitative Comparison

Finally, we compare τGAN to InterFaceGAN [24] and GANSpace [15] for the application of semantic face editing by using rotation transfer as one example.

Since the BU-3DFE database [33], see Sec. IV-A, contains 5000 faces images, 2500 from the left and from the 2500 right; we chose one of the two views as the reference image, and then used InterFaceGAN, GANSpace and τGAN to estimate a reconstruction of the image from the complementary rotation. The resulting image was then compared to the Ground Truth (GT) by 1) Pearson correlation coefficient (pcorr), 2) Structural Similarity Index Measure (SSIM) [31], and 3) Learned Perceptual Image Patch Similarity (LPIPS) [34]. For the LPIPS measure, we employed two versions: one based on VGG [26], referred to as lpips-vgg, and the other, lpips-alex, on AlexNet [20].

In InterFaceGAN [24] the authors find semantic directions

of StyleGAN by fitting SVMs to single semantic attributes using an annotated data set. Using these directions, semantic editing can be performed by interpolating in the direction $\mathbf{n} \in \mathbb{R}^N$ defined by the SVM hyper-plane normal vector for a given latent code $\mathbf{w} \in \mathcal{W}$, as

$$\mathbf{w}_{\text{edit}} = \mathbf{w} + \alpha \mathbf{n}, \tag{12}$$

where α is the strength of the shift in semantic direction associated with $\mathbf{n}$. To perform rotation transfer, we chose the pose direction for the StyleGAN1 model trained on FFHQ provided by [24] as $\mathbf{n}$.

GANSpace finds semantic directions in an unsupervised fashion using PCA. The semantic meaning of the found principal components needs to be assigned by a one-time manual labeling. In the paper the authors report that the 10th principal component applied only to the first 7 layers produces a shift in rotation for the pretrained StyleGAN1 network. Using this definition, and the rotation direction, we can perform semantic edits with GANSpace in a similar way as in eq. 12.

To determine the optimal interpolation strength α for both methods, we design an experiment where we perform rotation transfer with varying values for α. From the latent code representing an image of one rotation, we edit the latent code towards the complementary rotation resulting in a latent vector $\mathbf{w}_{\text{edit}}$ which is then used to synthesize an edited image. We then compare the edited image to the ground truth using the four metrics mentioned above. For each value of α we average the metrics and pick the minimum. The results are presented in Fig. 11, where it can be seen that the best performance for InterFaceGAN is reached at $\alpha = 2.77$, and for GANSpace at $\alpha = 1.66$, respectively. These values are used for the quantitative comparison presented in Fig. 13.

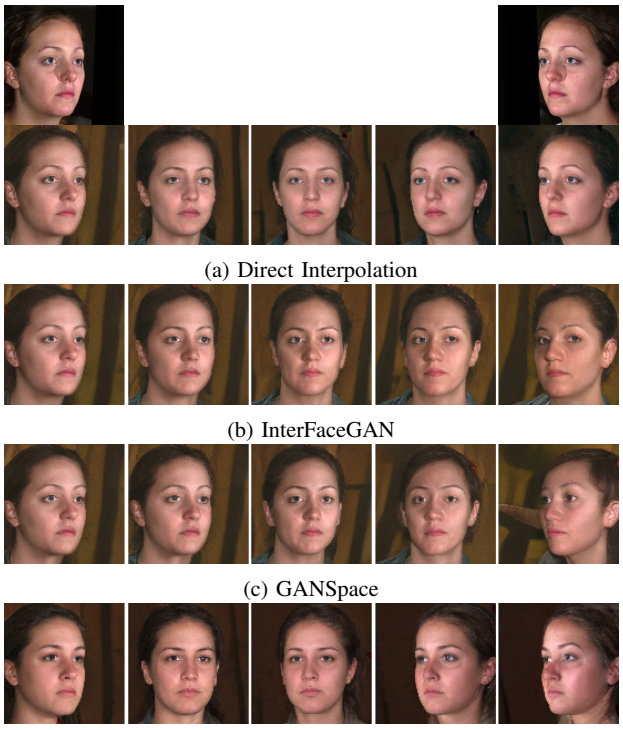
To perform rotation transfer with τGAN model, we first estimated the model parameter vectors $\widehat{\mathbf{q}}_k$, $k = 2, 3, 4$ for a given input image as described in Sec. III-D. Then we used the rotation subspace defined by $\mathbf{U}_4$ in (1). For τGAN we take the subspace direction $\mathbf{m} = \mathbf{u}_2^{(4)} - \mathbf{u}_1^{(4)} \in \mathcal{Q}_{\text{R}}$, where $\mathbf{u}_1^{(4)}$, $\mathbf{u}_2^{(4)}$ are the first and second row of $\mathbf{U}_4$, respectively. The rotation parameter was then changed as

$$\widetilde{\mathbf{q}}_4 = \widehat{\mathbf{q}}_4 + \gamma \mathbf{m}, \tag{13}$$

which then yields the edited latent code

$$\mathbf{w}_{\tau,\text{edit}} = \tau(\widehat{\mathbf{q}}_2 \otimes \widehat{\mathbf{q}}_3 \otimes \widetilde{\mathbf{q}}_4). \tag{14}$$

Fig. 12 shows synthesized images produced by InterFaceGAN, GANSpace and τGAN, respectively. These are compared against the reconstructions generated by latent codes interpolated directly in $\mathcal{W}$ space by $\mathbf{w} = \beta \mathbf{w}_{\text{left}} + (1 - \beta)\mathbf{w}_{\text{right}}$ where $\mathbf{w}_{\text{left}}$ and $\mathbf{w}_{\text{right}}$ refer to the left and right rotation, respectively. The results show that τGAN provides an alternative way for generating rotation in the StyleGAN latent space. Compared to InterFaceGAN, our model seems to create rotations which better preserve features like skin tone and gaze direction, and compared to GANSpace the face shape seems better preserved. However, for all methods we note that the identity of the person slightly changes in this example.



(a) Direct Interpolation

(b) InterFaceGAN

(c) GANSpace

(d) τGAN (Ours)

Fig. 12: Comparison of rotation transfer among varying methods. The ground truth images in pixel space are shown in the top row in the outermost columns. We use the latent code corresponding to the left hand rotation (top left) and try to recover the right hand rotation (top right). The provided images have been created by: (a) direct interpolation, (b) InterFaceGAN, (c) GANSpace, and (d) our proposed τGAN.

Additionally, we objectively compare the quality of rotation transfer resulting from different methods as follows. We apply the previously introduced three methods: InterFaceGAN, GANSpace, and our proposed τGAN, to shift the rotation of the 125 left-oriented images in the validation set towards the right orientation. We then compare the edited images to the known ground truth using the same four metrics introduced at the beginning of this section. The results in Fig. 13 show that τGAN has the lowest median value for all metrics when compared with InterFaceGAN and GANSpace.

V. Conclusions

In this work, we proposed τGAN, a tensor-based model for the auxiliary latent space of the StyleGAN. It is constructed by first embedding the images of the BU-3DFE database into the latent space of StyleGAN. The latent codes were stored into a tensor which is then factorized into semantically meaningful subspaces by HOSVD. This construction ensured that the semantic directions were directly interpretable in contrast to unsupervised methods, where this

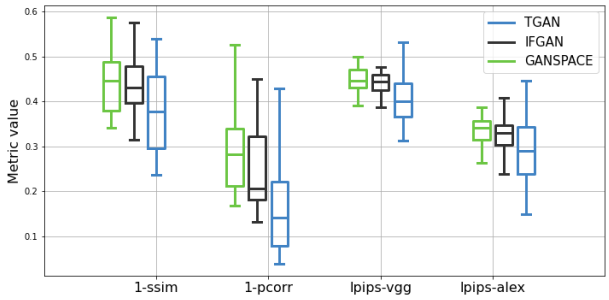


Fig. 13: Quantitative comparison of rotation transfer performed by varying methods. We start with images from the left rotation and shift the latent codes towards the right rotations using τGAN, InterFaceGAN, and GANSpace. The edited images are then compared to the GT based on the previously used adapted metrics, redefined to be the lower the better. We observe that the edited images produced by τGAN are more similar to the GT across all four metrics.

is not always the case.

We were able to generalize previous results [11] of face analysis by showing that the expression subspace has the structure where the expression trajectories meet in a specific *apathetic* expression, which is different from the mean or neutral face. We evaluated our approach quantitatively and qualitatively, and compared different versions of the proposed tensor models on the basis of approximation of unseen samples, and demonstrated the stability in the transfer of expression and rotation. From the results, we conclude that the proposed approach is a powerful way for characterizing and parameterizing the latent space of StyleGAN.

The current setting assumes complete data that contains measurements of all the people performing the same expressions from each rotation without any missing data. This requirement could be relaxed by low-rank completion methods that is left for future work. To conclude we employed a model trained on FFHQ, and received promising results on the BU-3DFE data set.

References

[1] R. Abdal, Y. Qin, and P. Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proc. ICCV*, pages 4431–4440, 2019.

[2] P. Baylies. Stylegan encoder - converts real images to latent space. https://github.com/pbaylies/stylegan-encoder/, 2019.

[3] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proc. SIGGRAPH*, pages 187–194, 1999.

[4] A. Brunton, T. Bolkart, and S. Wuhrer. Multilinear Wavelets: A Statistical Shape Space for Human Faces. In *Proc. ECCV*, pages 297–312, Jan. 2014.

[5] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. FaceWarehouse: A 3D Facial Expression Database for Visual Computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, Mar. 2014.

[6] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *NIPS*, pages 2172–2180, 2016.

[7] L. De Lathauwer and B. De Moor. A multi-linear singular value decomposition. *Society for Industrial and Applied Mathematics*, 21:1253–1278, 03 2000.

[8] Y. Deng, J. Yang, D. Chen, F. Wen, and X. Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *IEEE Computer Vision and Pattern Recognition*, 2020.

[9] C. Ferrari, G. Lisanti, S. Berretti, and A. D. Bimbo. A dictionary learning-based 3d morphable shape model. *IEEE Transactions on Multimedia*, 19(12):2666–2679, 2017.

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[11] S. Graßhof, H. Ackermann, S. Brandt, and J. Ostermann. Apathy is the root of all expressions. *12th IEEE Conference on Automatic Face and Gesture Recognition (FG2017)*, 2017.

[12] S. Graßhof, H. Ackermann, S. S. Brandt, and J. Ostermann. Multilinear Modelling of Faces and Expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3540–3554, Oct. 2021. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *Proc. CVPR*, pages 770–778, 2016.

[14] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proc. ICCV*, 2017.

[15] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris. Ganspace: Discovering interpretable gan controls. In *Proc. NeurIPS*, 2020.

[16] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020.

[17] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*, pages 4396–4405, 2019.

[18] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020.

[19] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM REVIEW*, 51(3):455–500, 2009.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105. Curran Associates Inc., 2012.

[21] D. Nikitko. Stylegan – encoder for official tensorflow implementation. https://github.com/puzer/stylegan-encoder/, 2019.

[22] D. Y. Park and K. H. Lee. Arbitrary style transfer with style-attentional networks. *In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:5873–5881, Dec 2018.

[23] S. Pidhorskyi, D. A. Adjeroh, and G. Doretto. Adversarial latent autoencoders. In *Proc. CVPR*, June 2020.

[24] Y. Shen, J. Gu, X. Tang, and B. Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020.

[25] Y. Shen and B. Zhou. Closed-form factorization of latent semantics in gans. In *CVPR*, 2021.

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[27] A. Tewari, M. Elgharib, G. Bharaj, F. Bernard, H.-P. Seidel, P. Pérez, M. Zöllhofer, and C. Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images, cvpr 2020. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, june 2020.

[28] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, ECCV '02, page 447–460, Berlin, Heidelberg, 2002. Springer-Verlag.

[29] M. A. O. Vasilescu and D. Terzopoulos. Multilinear subspace analysis of image ensembles. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages II– 93, 07 2003.

[30] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face Transfer with Multilinear Models. In *ACM SIGGRAPH*, pages 426–433, 2005.

[31] Z. Wang and A. Bovik. A universal image quality index. *Signal Processing Letters, IEEE*, 9:81 – 84, 04 2002.

[32] K. Yano and K. Harada. Multilinear face model. In *Visualization, Imaging, and Image Processing (VIIP 2008)*, 2008.

[33] L. Yin, X. Wei, Y. Sun, J. Wang, and M. Rosato. A 3d facial expression database for facial behavior research. In *7th Intern. Conf. on Automatic Face and Gesture Recognition (FGR06)*, pages 211–216, 2006.

[34] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

A.2 Paper II:
Tensor-based Emotion Editing in the StyleGAN Latent Space

Tensor-based Emotion Editing in the StyleGAN Latent Space

René Haas, Stella Graßhof, and Sami S. Brandt
IT University of Copenhagen
{renha,stgr,sambr}@itu.dk

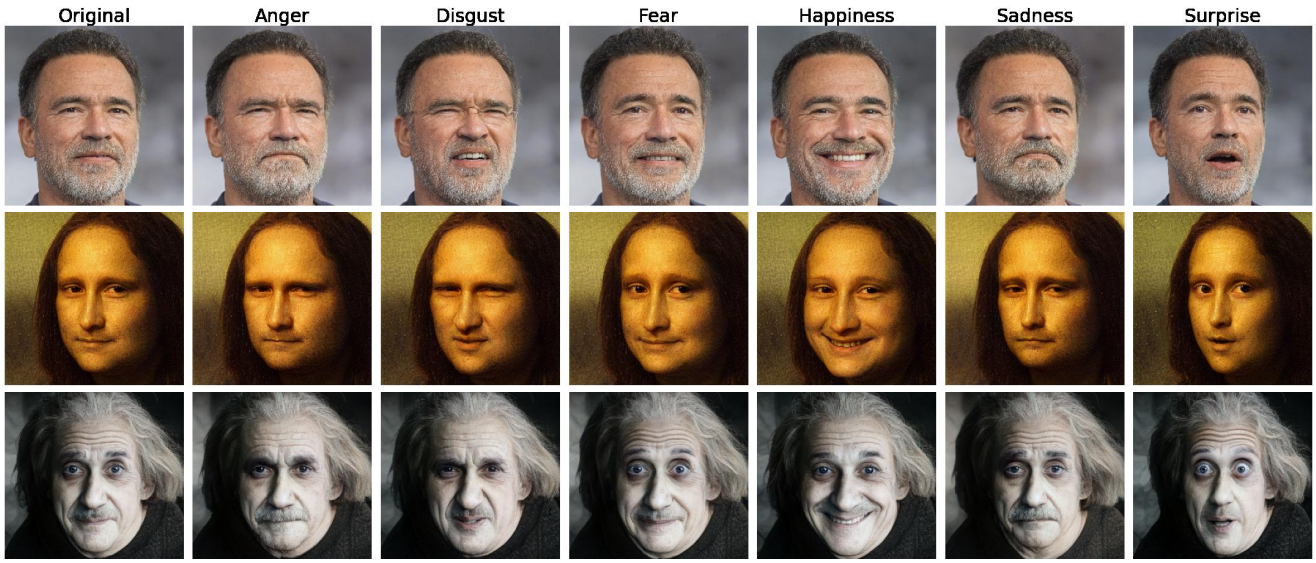


Figure 1. Using our model we can edit StyleGAN latent codes in the direction of the six prototypical emotions.

Abstract

In this paper, we use a tensor model based on the Higher-Order Singular Value Decomposition (HOSVD) to discover semantic directions in Generative Adversarial Networks. This is achieved by first embedding a structured facial expression database into the latent space using the e4e encoder. Specifically, we discover directions in latent space corresponding to the six prototypical emotions: anger, disgust, fear, happiness, sadness, and surprise, as well as a direction for yaw rotation. These latent space directions are employed to change the expression or yaw rotation of real face images. We compare our found directions to similar directions found by two other methods. The results show that the visual quality of the resultant edits are on par with State-of-the-Art. It can also be concluded that the tensor-based model is well suited for emotion and yaw editing, i.e., that the emotion or yaw rotation of a novel face image can be robustly changed without a significant effect on identity or other attributes in the images.

1. Introduction

Generative Adversarial Networks (GANs) [12] have emerged as one of the most promising architectures for image synthesis. GANs can produce synthetic images with near-perfect photorealism [5, 18–21]. GANs learn to organize the data they are trained on into a latent space and are, by drawing samples from the latent space, able to synthesize new images which are not contained in the training data but follow the same distribution. In particular, in the field of face synthesis StyleGAN has set new standards for what is possible [19–21].

Recent work has explored methods to gain artistic control over the images produced by modern GANs [1, 17, 25, 29, 33–35, 41]. In this work, we use a multilinear tensor model to derive latent space directions in StyleGAN2 [21] corresponding to the six prototypical emotions: anger, disgust, happiness, fear, sadness, and surprise as well as yaw rotation. With these directions, we are able to edit the emotion of real face images as shown in Fig. 1.

StyleGAN. The StyleGAN generator G is composed of two networks, the *mapping network* f and the *synthesis network* g. The mapping network f maps the latent vector $\mathbf{z} \in \mathcal{Z}$ onto the auxiliary latent space $\mathcal{W}$ while the synthesis network maps a vector $\mathbf{w} \in \mathcal{W}$ to the final output image. The latent vectors in $\mathcal{Z}$ follow the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ while the distribution of the auxiliary latent codes in $\mathcal{W}$ is learned by the mapping network f. The main benefit of this mapping is that the $\mathcal{W}$ space is more disentangled if compared to the $\mathcal{Z}$ space [20].

Every major block corresponding to a resolution of the synthesis network is modulated by two style vectors $w_1, w_2 \in \mathbb{R}^{512}$. Thus, for the full 1024 by 1024 generator there are 9 major blocks and the synthesis network takes a total of 18 style vectors as an input. Each set of style vectors has different effects on the synthesized image. In detail, the style vectors for the early layers, corresponding to coarse spatial resolutions, control high-level aspects of the image such as pose and face shape. Style vectors on the middle layers control smaller scale facial features like hair style and if the eyes and mouth are open or closed. The style vectors on the later layers correspond to higher resolutions controls such as the texture and the microstructure of the generated image [20]. In $\mathcal{W}$ space, each of the style vectors are identical. However, we can also allow them to be different, in which case the resulting space is denoted as the $\mathcal{W}+$ space. The $\mathcal{W}+$ space can be used for style mixing [20] and GAN inversion [28, 45]. Recently, an additional latent space referred to as *style space* has also been proposed [41].

Semantic Face Editing. Several methods have been proposed to enable edits of the images produced by StyleGAN. InterFaceGAN [32, 33] uses pre-trained binary classifiers to annotate StyleGAN generated images based on single binary attributes, e.g., young vs. old, male vs. female, glasses vs. no glasses. Support vector machines are then trained on the annotated data to discriminate between each attribute in the latent space. The normal vectors of the separating hyperplane define a direction in latent space that changes the corresponding binary attribute. GANSpace [17] finds interpretable directions in an unsupervised fashion with PCA while manual examination of the found directions is required. Directions found with PCA are typically entangled, affecting multiple attributes. It was shown that the degree of entanglement can be reduced by only applying the found directions to a subset of the style vectors. It has also been proposed to make the eigenvalue decomposition on the weights of the pre-trained generator to discover meaningful semantic directions in the latent space [34]. Recently, StyleCLIP [25] demonstrates text driven semantic editing by minimizing CLIP [27] loss between a text input and the generated image. StyleFlow [1] proposed editing along non-linear paths using normalizing flows to better preserve identity.

Separate from StyleGAN research, different multilinear methods have been widely used to model and analyze faces and expressions [4, 10, 15, 38]. Recently there has been some interest in applying these methods to explore the latent space of GANs. For example, StyleRig [35] proposes edits by minimizing the loss between the image produced by the generated image and an image rendered by a 3D morphable model. Furthermore, models based on the Higher-Order Singular Value Decomposition (HOSVD) have successfully been used to model faces, their 3D reconstruction, as well as in transferring expressions [6, 7, 39, 40]. Recently, it has been suggested [16] to use such a HOSVD-based tensor model for semantic face editing in StyleGAN. Here a facial expression database was projected into the StyleGAN $\mathcal{W}+$ space and relevant semantic subspaces corresponding to identity, expression and yaw rotation were defined using HOSVD-based subspace factorization. The model showed limited flexibility for representing arbitrary latent codes and to overcome this a stacked style-separated model was proposed. This extended the tensor model to an ensemble of tensor models, one for each style vector in the StyleGAN $\mathcal{W}+$ space. Further, it was shown that in the derived expression subspace, each of the six prototypical emotions formed nearly linear trajectories in agreement with [14]. Although initial results were promising, convincing expression editing using a HOSVD-based model on the StyleGAN latent space was however not yet demonstrated. We propose a solution to this shortcoming, and demonstrate the robustness, and competitiveness of our approach in this work.

Generator Inversion. To facilitate editing of real images, the images first need to be projected into the StyleGAN latent space. This is also referred to as GAN inversion [46] and the problem is to find a latent code that, when passed to the generator, produces an image as close as possible to the given target image. Typically GAN inversion techniques are either based on training an encoder [2, 26, 30, 37], which can embed an image into latent space at inference time, or optimization-based techniques [21, 28, 29, 42]. In the latter approach, the latent code is found by minimizing a loss function, typically pixel-wise L2 or perceptional image similarity [44] is used. Hybrid approaches have also been proposed which use a trained encoder to find a good initial condition for subsequent iterative optimization of the latent code [23, 45].

Recently, [31] shows that novel images can be embedded into $\mathcal{W}$ space with a lower reconstruction error by fine-tuning the pre-trained generator on the target image such that the latent code in $\mathcal{W}$ space yields an image closer to the target.

Recent work [37] suggests that there is a trade-off between distortion and editability when selecting which latent

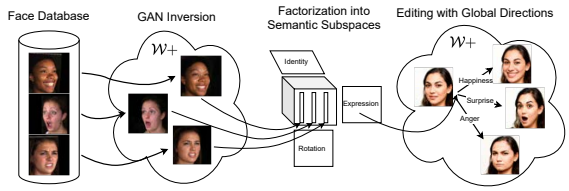


Figure 2. Diagram of our method. We first project a facial expression database intro the $\mathcal{W}+$ space of StyleGAN. We then use the HOSVD to factorize the latent representation of the data in order to derive meaningful semantic subspaces. From the subspaces we define a set of global editing directions in $\mathcal{W}+$ corresponding to yaw rotation and each of the six basic emotions.

space to project a given target image into. When projecting out-of-domain images into the StyleGAN latent space picking the extended $\mathcal{W}+$ space leads to a higher quality reconstruction, i.e, it yields an image closer to the target image. However, latent codes in the $\mathcal{W}+$ space are generally less editable than latent codes in $\mathcal{W}$ space. To find latent codes with the optimal trade-off between distortion and editability a novel training methodology was proposed [37] which embeds images into $\mathcal{W}+$ space in a way that constrains the latent codes to be as close to $\mathcal{W}$ space as possible.

Contributions. Our contributions can be summarized as follows

- We show that a HOSVD-based tensor model is able to discover novel semantic directions robustly, corresponding to the six prototypical emotions, in pretrained GANs.
- We show that convincing emotion directions can be derived by truncating the expression intensity subspace.
- We show that, by using the e4e encoder [37] for projecting real images into the latent space of StyleGAN, it is possible to construct a tensor model which enables stable rotation and expression transfer on real faces.
- We show the previously proposed tensor model for the GAN latent space [16] had an implicit rank-one constraint, which can be relaxed, leading to lower reconstruction error.

2. Method

In this section, we describe tensor model formulation [16] and propose two extensions to it: (1) We show how to relax the implicit rank-one constraint of the model by replacing the set of parameter vectors of the model with a single full rank parameter tensor, and (2) show how to derive emotion directions in $\mathcal{W}+$ by truncating the expression intensity subspace. An overview of our approach is shown in Fig. 2.

2.1. Multilinear Tensor Model

Given a data set of StyleGAN latent codes in $\mathcal{W}+$ we represent them so that each latent code is equivalent to a vector $\mathbf{w} \in \mathbb{R}^D$, where $D = 9216$ for the generator producing 1024×1024 images. Suppose we have latent codes for P different persons, performing E expressions each with I different intensities from R different rotations, then we arrange the data into the 5[th] order tensor $T \in \mathbb{R}^{D \times P \times E \times I \times R}$. We then proceed to calculate the Higher-Order Singular Value Decomposition (HOSVD) on the mean-centered data tensor as

$$T - \bar{T} = S \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3 \times_4 \mathbf{U}_4 \times_5 \mathbf{U}_5, \quad (1)$$

where S is the core tensor and $\times_n$ denotes the n-mode tensor matrix product. The mean tensor is written as $\bar{T} = \bar{\mathbf{w}} \otimes \mathbf{1}_P \otimes \mathbf{1}_E \otimes \mathbf{1}_I \otimes \mathbf{1}_R$, where $\bar{\mathbf{w}}$ is the mean latent code from the data set, $\mathbf{1}_P$ is a vector of ones with dimension P, and $\otimes$ denotes the tensor product. The $\mathbf{U}_i$ matrices have orthonormal columns, i.e., $\mathbf{U}_i^{\mathrm{T}}\mathbf{U}_i = \mathbf{I}$ and are constructed from the left singular vectors of the mode-n matrix unfoldings of the mean-centered data tensor. The columns of $\mathbf{U}_i$ form the basis for the respective subspace. The columns of $\mathbf{U}_1$ form a basis for the latent space and are identical to the principal components [15]. Likewise $\mathbf{U}_2$, $\mathbf{U}_3$, $\mathbf{U}_4$, and $\mathbf{U}_4$ form the bases for the person identity, expression, intensity and rotation subspaces respectively.

Parameter Vectors. To recover a specific latent code from the tensor model, we select appropriate rows of $\mathbf{U}_2$, $\mathbf{U}_3$, $\mathbf{U}_4$ and $\mathbf{U}_5$ corresponding to the desired person, expression, expression intensity, and rotation respectively. By introducing one-hot vectors $\mathbf{q}'_i$ which we will refer to as the *canonical* parameters for the tensor model, we get

$$\hat{\mathbf{w}} = \bar{\mathbf{w}} + C \times_2 \mathbf{q}'^{\mathrm{T}}_2\mathbf{U}_2 \times_3 \mathbf{q}'^{\mathrm{T}}_3\mathbf{U}_3 \times_4 \mathbf{q}'^{\mathrm{T}}_4\mathbf{U}_4 \times_5 \mathbf{q}'^{\mathrm{T}}_5\mathbf{U}_5, \quad (2)$$

where $C = S \times_1 \mathbf{U}_1$. This formulation is analogous to the one proposed in [14, 15] and subsequently, [16]. Now, (2) can be further simplified by defining $\mathbf{q}_i^{\mathrm{T}} = \mathbf{q}'^{\mathrm{T}}_i\mathbf{U}_i$ which allows us to write

$$\hat{\mathbf{w}} = \bar{\mathbf{w}} + C \times_2 \mathbf{q}_2^{\mathrm{T}} \times_3 \mathbf{q}_3^{\mathrm{T}} \times_4 \mathbf{q}_4^{\mathrm{T}} \times_5 \mathbf{q}_5^{\mathrm{T}}, \quad (3)$$

which gives is a more compact representation of the tensor model.

Recovering Subspace Parameters. To find the parameters $(\mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5)$ for a novel latent code $\mathbf{w}$, with corresponding to the latent code $\hat{\mathbf{w}}$ which best approximates $\mathbf{w}$, one could minimize the L_2 loss,

$$\mathcal{L}(\mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5) = ||\hat{\mathbf{w}}(\mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5) - \mathbf{w}||_2^2. \quad (4)$$

Additionally, it has been proposed in [14] to regularize the solution by the Tikhonov regularizer and sum constraint as

$$\mathcal{R}(\mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5) = \sum_{i=2}^{5} \left[\lambda_{1,i} ||\mathbf{q'}_i^{\mathrm{T}}||_2^2 + \lambda_{2,i} (\mathbf{q'}_i^{\mathrm{T}} \mathbf{1} - 1)^2 \right], \tag{5}$$

that yields the regularized minimization problem

$$\min_{\mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5} \mathcal{L}(\mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5) + \mathcal{R}(\mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5). \tag{6}$$

This regularization is important for finding a stable parameter vector representations and thereby enables expression editing for latent codes corresponding to novel images, as will be seen below.

Relaxing the Rank-One Constraint. In the tensor model (3), each latent code is entirely determined by four parameter vectors $\mathbf{q}_2$, $\mathbf{q}_3$, $\mathbf{q}_4$ and $\mathbf{q}_5$ corresponding to identity, expression, expression intensity and rotation, respectively. Using component notation and the Einstein summation convention we rewrite (3) as

$$\widehat{w}_i = \bar{w}_i + C_{ijklm} q_j^{(2)} q_k^{(3)} q_l^{(4)} q_m^{(5)}, \tag{7}$$

where $Q_{jklm} = q_j^{(2)} q_k^{(3)} q_l^{(4)} q_m^{(5)}$ is a rank-one tensor.

Now, we propose to relax this implicit rank-one constraint and instead allow the tensor Q_{jkl} to be full rank that leads to the problem

$$\min_{Q} ||\widehat{\mathbf{w}}(Q) - \mathbf{w}||_2^2. \tag{8}$$

The relaxation increases the number of parameters of the tensor model from $P + E + I + R$ parameters to $PEIR$ parameters. This results in a more flexible model which yields lower reconstruction errors for novel latent codes.

2.2. Truncating the Expression Intensity Subspace

From (1), the expression intensity subspace is truncated to a one-dimensional subspace by selecting the dominant singular vector, i.e., the first column of $\mathbf{U}_4$ which we denote $\tilde{\mathbf{u}}_4$. The truncated core tensor is then written as

$$\widetilde{S} = (T - \bar{T}) \times_1 \mathbf{U}_1^{\mathrm{T}} \times_2 \mathbf{U}_2^{\mathrm{T}} \times_3 \mathbf{U}_3^{\mathrm{T}} \times_4 \tilde{\mathbf{u}}_4^{\mathrm{T}} \times_5 \mathbf{U}_5^{\mathrm{T}}. \tag{9}$$

Defining $\widetilde{C} = \widetilde{S} \times_1 \mathbf{U}_1$ as before, then the model is written similarly to (2) and (3) as

$$\widehat{\mathbf{w}} = \bar{\mathbf{w}} + \widetilde{C} \times_2 \mathbf{q'}_2^{\mathrm{T}} \mathbf{U}_2 \times_3 \mathbf{q'}_3^{\mathrm{T}} \mathbf{U}_3 \times_4 \mathbf{q'}_4^{\mathrm{T}} \tilde{\mathbf{u}}_4 \times_5 \mathbf{q'}_5^{\mathrm{T}} \mathbf{U}_5, \tag{10}$$

where the corresponding intensity parameter $\mathbf{q'}_4^{\mathrm{T}} \tilde{\mathbf{u}}_4 = q_4$ is a scalar since the expression intensity subspace has been truncated. Thus, the expression intensity factors out of the model and we may write

$$\widehat{\mathbf{w}} = \bar{\mathbf{w}} + q_4 (\widetilde{C} \times_2 \mathbf{q}_2^{\mathrm{T}} \times_3 \mathbf{q}_3^{\mathrm{T}} \times_5 \mathbf{q}_5^{\mathrm{T}}), \tag{11}$$

where q_4 can now be interpreted as the expression intensity parameter. We trivially unfold the singleton dimension of $\widetilde{C}$ corresponding to the intensity subspace, i.e., $\widetilde{C}_{ijklm} \rightarrow \widetilde{C}_{ijkm}$ and then write the model as

$$\widehat{w}_i = \bar{w}_i + q^{(4)} \widetilde{C}_{ijkm} q_j^{(2)} q_k^{(3)} q_m^{(5)}. \tag{12}$$

2.3. Recovering Semantic Directions

Emotion Directions. We define emotion directions in latent space by selecting an appropriate row $\mathbf{q}_3^{\text{expr}}$ of $\mathbf{U}_3$ corresponding to the emotion of interest. The combined parameter tensor corresponding to an expression direction is then written as

$$Q^{(\text{expr})} = \bar{\mathbf{q}}_2 \otimes \mathbf{q}_3^{\text{expr}} \otimes \bar{\mathbf{q}}_5, \tag{13}$$

where $\bar{\mathbf{q}}_2$ and $\bar{\mathbf{q}}_5$ is the mean person and rotation parameters respectively. To change the expression of a given latent code $\mathbf{w}$, we interpolate linearly in the direction given by the vector $\mathbf{n}^{(\text{expr})}$ with components

$$n_i^{(\text{expr})} = \widetilde{C}_{ijkm} Q_{jkm}^{(\text{expr})}, \tag{14}$$

thus performing an expression edit as

$$\mathbf{w}_{\text{edit}}^{(\text{expr})} = \mathbf{w} + q_4 \mathbf{n}^{(\text{expr})}. \tag{15}$$

Rotation Direction. We edit rotations in a similar way. First we select the mean person, expression and expression intensity parameters $\bar{\mathbf{q}}_2$ $\bar{\mathbf{q}}_3$ and $\bar{q}_4$ and then define the rotation direction parameter $\mathbf{q}_5^{(\text{rot})}$ as the difference between the parameters corresponding to the left and right rotations, i.e., the difference between the two rows of $\mathbf{U}_5$. We write the rotation direction parameter directly as

$$\mathbf{q}_5^{(\text{rot})} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}^{\mathrm{T}} \mathbf{U}_5. \tag{16}$$

Now the combined rotation direction tensor is written as

$$Q^{(\text{rot})} = \bar{q}_4 (\bar{\mathbf{q}}_2 \otimes \bar{\mathbf{q}}_3 \otimes \mathbf{q}_5^{(\text{rot})}), \tag{17}$$

and we can change the rotation of a latent code as

$$\mathbf{w}_{\text{edit}}^{(\text{rot})} = \mathbf{w} + \beta \mathbf{n}^{(\text{rot})} \quad \text{with} \quad n_i^{(\text{rot})} = \widetilde{C}_{ijkm} Q_{jkm}^{(\text{rot})}, \tag{18}$$

where β is the strength of the rotation.

With this formulation, we apply semantic edits directly in $\mathcal{W}+$ without the need for estimating the tensor model parameters beforehand as has otherwise been suggested [16].

(a) (b) (c)

Figure 3. Image embeddings. (a) BU-3DFE images, (b) random samples from the generator, and (c) real images. The embeddings of the original images are shown in the top row, the parameter vector embeddings in the middle, and the parameter tensor embeddings in the bottom row.

3. Experiments

Our tensor model was trained with the latent space projection of images from the Binghamton University 3D Facial Expression database (BU-3DFE) [43]. The BU-3DFE database contains 2500 3D face scans and corresponding images from two views of 100 persons (56 female and 44 male) with varying ages (18-70 years), and diverse ethnic/racial ancestries. Each subject was asked to perform the six basic emotions: anger, disgust, fear, happiness, sadness, and surprise, each with four levels of intensity. Additionally, for each participant, a neutral face is provided. Hence, for each person, there are 25 facial expressions in total, recorded from two pose directions, left and right, resulting in 5000 face images. Additionally, we used the FEI face database [36] which contains 14 images of each of the 200 individuals, 100 male and 100 female. For each the database contains two frontal images, one with a neutral or non-smiling expression and the other with a smiling facial expression, the rest of the images depicts each individual with a neutral expression from various yaw rotations.

3.1. Implementation Details

We use the full resolution, i.e. 1024×1024, StyleGAN2 [19] generator which has been pre-trained on FFHQ [20]. The tensor model was implemented in PyTorch [24] using tntorch [3] to calculate the HOSVD. To estimate the tensor model parameters we used gradient descent implemented in PyTorch with the Adam optimizer. For comparing images we use two different metrics. For perceptual image similarity we use LPIPS [44] and for identity similarity we uses Arcface [9]. To measure the pose of the generated images we uses MediaPipe [22] to extract 2D and 3D landmarks and then proceeded to solve the Perspective-n-point (PnP) [11] problem which gave us a scalar value for the yaw rotation of a given image. We embedded all images into $\mathcal{W}+$ space using the e4e encoder [37].

Table 1. Comparison of reconstruction error $||\widehat{\mathbf{w}} - \mathbf{w}||_2^2$ by representing randomly sampled latent codes and latent codes from the BU-3DFE data set with parameter vector and a parameter tensor respectively.

	Random Latents	BU-3DFE Latents
Rank one	$(12 \pm 3) \times 10^2$	$(1.7 \pm 0.2) \times 10^2$
Full rank	$(6 \pm 1) \times 10^2$	7 ± 1

3.2. Subspace Parameter Recovery

We computed estimated the tensor model parameters for 3 types of novel latent codes: 1) BU-3DFE latent codes where we left one person out in the calculation of the tensor model, 2) randomly sampled latent codes, and 3) real images projected into latent space. Fig. 12 shows the result of recovering the tensor model parameters for these three types of latent codes when recovering the parameters in vector and tensor form, respectively. It can be seen that using parameter vectors for the tensor model led to a significant reconstruction loss if compared to using a representation with a parameter tensor, as illustrated in Fig. 4 and quantified in Tab. 1. It seems that the randomly sampled images are slightly harder to reconstruct than the embedded real images.

For the representation with parameter vectors, we find that although the proposed regularization (5) leads to a slightly higher reconstruction error, it is important in order to find parameter vectors which are suitable for expression editing. Fig. 5 shows that performing expression edits on the regularized parameters leads to less identity change compared to the non-regularized parameters. The importance of regularization is more noticeable when we recover the parameters for a randomly generated image if compared to an image contained the in BU-3DFE database.

Moreover, it can be seen that the tensor model is not necessary for expression editing, because we can edit the latent

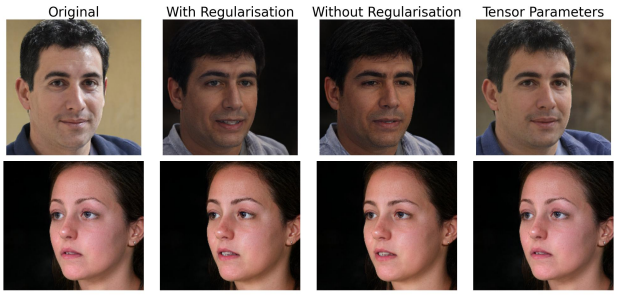
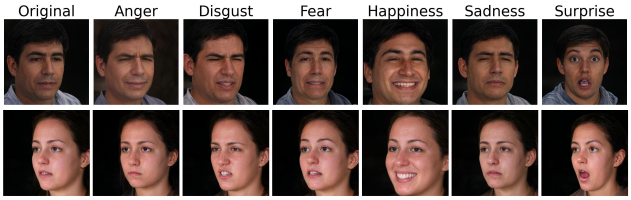


Figure 4. Representing a latent code in the tensor model with parameter vectors with and without regularization compared with a representation using a parameter tensor.

(a) Without regularization.



(b) With regularization.

Figure 5. Visual comparison of the effect of regularization for expression editing using parameter vectors for the tensor model.

Figure 6. Direct edit in the $\mathcal{W}+$ space without prior estimation of the model parameters.

code directly by perturbing in the directions defined by (15), instead of manipulating the estimated parameters of the tensor model. The effect of such a direct edit is illustrated in Fig. 6. The main advantage of performing expression edits in this way, is that we avoid the reconstruction error associated with representing the latent code in terms of the tensor model parameters.

3.3. Expression Direction Recovery

Fig. 7 shows the effect of applying the found six latent space directions to the BU-3DFE mean face. We found that subtracting the sadness direction from the mean face also produces a happy facial expression. However, the resulting expression is qualitatively different from adding the happy direction to the mean face. While adding the happy direction results in a wide smile, subtracting the sadness direction results in a smile that is narrower but where the mouth is more open. See the supplementary materials for videos showing the found emotion directions on real face images.

3.4. Comparison to Related Work

We compared the rotation and smile directions found by our approach to those previously found by InterFaceGAN [33] and GANSpace [17]. For InterFaceGAN, we used the PyTorch version of the rotation and smile directions provided by the authors of [31] at their GitHub repository[1]. For the rotations, we chose a manipulation strength that resulted in a similar degree of rotation. To perform rotations with GANSpace [17], we initially used the 2nd principal component applied to the first three style vectors. However, we found that if we only changed the first three style

[1] https://github.com/danielroich/PTI/tree/main/editings/interfacegan_directions

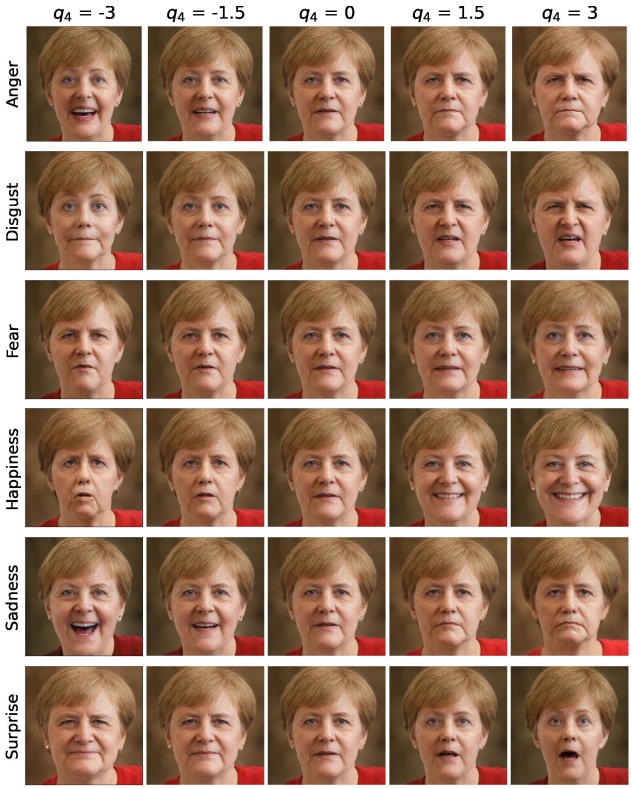


Figure 7. Effect of applying the direction corresponding to the six prototypical expressions to a real image. The rows show the different expressions determined by $\mathbf{q}_3$ while the strength is modulated by q_4, while the rotation parameters $\mathbf{q}_5$ remain unchanged. The right column shows edits in the direction of the respective expression while the left column illustrates the subtraction of it.

Figure 8. Comparison of rotations produced by GANSpace [17] (top 2 rows), InterFaceGAN [33] (third row) and our approach (bottom). Here GANSpace* refers to a manipulation where we edit the first five style vectors rather than the first three as described in the main text.

vectors to edit the rotation, the result tends to break down when the editing strength is large, which is demonstrated in the first row in Fig. 8. If we applied the edit to the first five style vectors instead, we generally received better results, see second row in Fig. 8.

We visually compared the rotations by GANSpace, InterFaceGAN and our proposed method on images which are randomly sampled from the generator as well as images from the FEI face database [36]. For the FEI database we used the frontal face images as initial conditions and then applied rotations with GANSpace, InterFaceGAN and our method to approximate the latent codes corresponding to rotated images from the database. The results on randomly sampled images are shown in Fig. 8 and on the FEI database in Fig. 9, respectively. It can be seen that the quality of the edits are visually on par, except the gaze direction follows the camera in the InterFaceGAN results.

3.5. Happy Faces

We compared the found happiness direction to the smile directions from GANSpace and InterFaceGAN, respectively. For GANSpace we used the 47^{th} principal component applied to the 5^{th} and 6^{th} style vectors. The results are shown in Fig. 10. Although each method resulted in a smile in the generated image, the style of smile is different. Our method yielded a wider smile whereas GANSpace yielded a smile with a larger mouth opening, while the smile by InterFaceGAN seems to fall between these two.

3.6. Face Frontalization

To experiment face frontalization, we started with the latent codes corresponding to the rotated images in the FEI database [36], then edited the yaw of latent code to frontalize the images. Quantative comparison is shown in Fig. 11. In Tab. 2, we compare the perceptual and identity similarity scores of the frontalized images to the ground truth. It can be seen the frontalized images are very similar to the result obtained by using the pose direction from InterFaceGAN. However, our method yielded better similarity scores against to the ground truth. In addition, the gaze direction by InterFaceGAN is not straight ahead whereas ours is.

Figure 9. Qualitative comparison of the found rotation direction with the equivalent edits from InterFaceGAN [33] and GANSpace [17] applied on the FEI face database [36].

Figure 10. Visual comparison of editing a randomly sampled latent code in the smiling directions found in GANSpace [17] and InterFaceGAN [33] with the happiness direction found in this work.

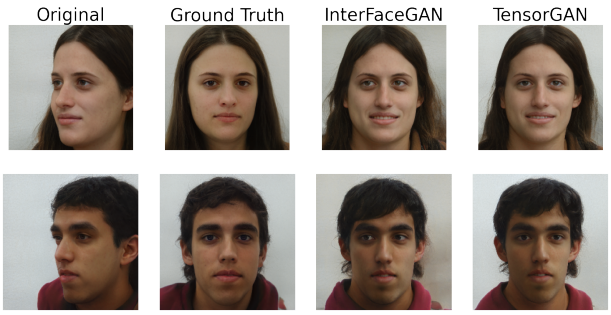


Figure 11. Qualitative comparison of facial frontalization with InterFaceGAN [33] and our method on FEI face database [10].

Table 2. Comparison of perceptual and identity similarity scores of facial frontalization of images from the FEI face database with InterFaceGAN [33] and our method. The results are reported as mean value $\pm$ standard error of the mean.

	LPIPS [44]	ArcFace [9]
InterFaceGAN	0.315 ± 0.003	0.402 ± 0.008
TensorGAN	0.305 ± 0.004	0.372 ± 0.008

3.7. Validation with expression classifier

To validate that the semantic directions recovered with our approach produce a change in the generated images corresponding to the intended labels, we use a pre-trained expression classifier [8] which is trained on the FER2013 data set [13]. We sampled 5×10^3 random images with varying expressions from StyleGAN and edited these in the direction of each basic emotion. Using the classifier, we obtained the probability mass distribution of expressions for the sampled and edited images. From this, we calculated the average difference in probability mass due to the edit and visualize the results with a heatmap in Fig. 12.

The edits in the direction of anger, happiness, sadness, and surprise lead to changes in the class probabilities which corresponds to an increase in probability of the expected class labels. However, the edits in the disgust direction lead to an increase in probability for anger as well as disgust while edits in the fear direction leads to a larger probability mass for the surprise label. This is explained by the fact that PyFeat also classifies the BU-3DFE raw images in a similar way as can be seen in the confusion matrix in Fig. 13. Thus, this discrepancy is not due to a limitation of our model, but rather due to systematic differences between the BU-3DFE and FER2013 data sets, which are especially apparent for data points annotated with the fear or disgust labels.

4. Conclusion

In this work, we have presented an extension of the HOSVD-based tensor model, proposed in [16]. In contrast to [16], (1) we use the e4e encoder [37] to recover highly editable latent codes for the BU-3DFE database, (2) we improve reconstruction in the tensor model by allowing the parameters to be full-rank, and (3) we show that edits can be applied directly in latent space. Further, we showed that we can calculate linear directions in latent space corresponding to the six prototypical emotions by truncating the emotion intensity subspace. After obtaining a latent representation of the data, constructing the tensor model is fast, requiring only a few minutes to calculate the HOSVD. Further, the latent space directions corresponding to the six prototypical emotions can be calculated from the tensor model and subsequently applied to any latent code in the original latent space without the need to first estimate the subspace parameters as otherwise suggested in [16]. In other words, the found semantic directions are global and can be applied to any latent code without any further calculations. Our method is able to identify directions in latent space corresponding to yaw rotation, as well as each of the six basic expressions. The quality of the edits performed with these directions is on par with the corresponding edits using GANSpace [17] and InterFaceGAN [33].

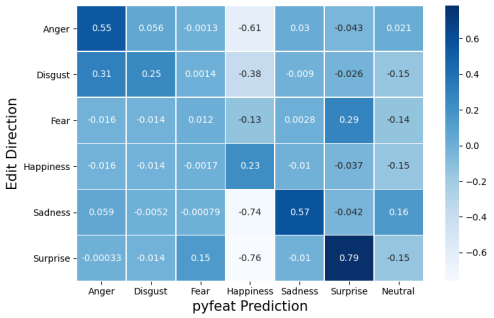


Figure 12. Heatmap of the average difference in expression probability masses due to expression edits with our approach. Note that Fear increases the probability mass for Surprise and Disgust increases the probability mass for Anger. The reason is explained in the main text.

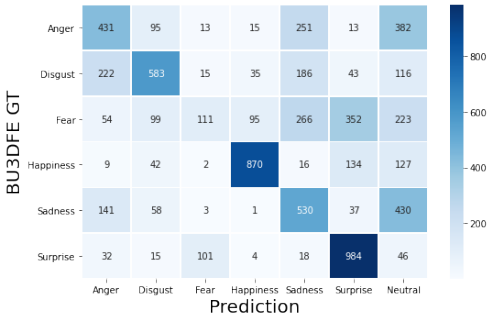


Figure 13. Confusion matrix showing the Pyfeat classification results on BU-3DFE. It shows that the correlation between Fear/Surprise and Disgust/Anger is not due to a limitation of our model, but can attributed to the differences between the BU-3DFE and FER2013 data sets.

References

[1] Rameen Abdal, Peihao Zhu, Niloy J. Mitra, and Peter Wonka. StyleFlow: Attribute-conditioned exploration of StyleGAN-generated images using conditional continuous normalizing flows. *ACM Trans. Graph.*, 40(3), May 2021. 1, 2

[2] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021. 2

[3] Rafael Ballester-Ripoll. *tntorch - Tensor Network Learning with PyTorch*. Oct 2021. 5

[4] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH*, pages 187–194, 1999. 2

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, Feb 2019. 1

[6] Alan Brunton, Timo Bolkart, and Stefanie Wuhrer. Multilinear wavelets: A statistical shape space for human faces. In *Proc. ECCV*, pages 297–312, 2014. 2

[7] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. FaceWarehouse: A 3D facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, Mar 2014. 2

[8] Jin Hyun Cheong, Tiankang Xie, Sophie Byrne, and Luke J Chang. Py-feat: Python facial expression analysis toolbox. page 25. 8

[9] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proc. CVPR*, pages 4690–4699, 2019. 5, 8

[10] Claudio Ferrari, Giuseppe Lisanti, Stefano Berretti, and Alberto Del Bimbo. A dictionary learning-based 3D morphable shape model. *IEEE Transactions on Multimedia*, 19(12):2666–2679, 2017. 2, 8

[11] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, 1981. 5

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, page 2672–2680. Curran Associates, Inc., 2014. 1

[13] Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, Radu Ionescu, Marius Popescu, Cristian Grozea, James Bergstra, Jingjing Xie, Lukasz Romaszko, Bing Xu, Zhang Chuang, and Yoshua Bengio. Challenges in Representation Learning: A report on three machine learning contests. *arXiv:1307.0414 [cs, stat]*, July 2013. arXiv: 1307.0414. 8

[14] Stella Graßhof, Hanno Ackermann, Sami Brandt, and Jörn Ostermann. Apathy is the root of all expressions. *12th IEEE Conference on Automatic Face and Gesture Recognition (FG2017)*, 2017. 2, 3, 4

[15] Stella Graßhof, Hanno Ackermann, Sami Sebastian Brandt, and Jörn Ostermann. Multilinear modelling of faces and expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3540–3554, Oct. 2021. 2, 3

[16] René Haas, Stella Graßhof, and Sami Sebastian Brandt. Tensor-based subspace factorization for StyleGAN. *arXiv:2111.04554 [cs]*, Nov 2021. arXiv: 2111.04554. 2, 3, 4, 8

[17] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. GANSpace: Discovering interpretable GAN controls. In *Proc. NeurIPS*, 2020. 1, 2, 6, 7, 8

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *Proc. ICLR*, Feb 2018. 1

[19] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020. 1, 5

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*, pages 4396–4405, 2019. 1, 2, 5

[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020. 1, 2

[22] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, and et al. MediaPipe: A framework for building perception pipelines. *arXiv:1906.08172 [cs]*, Jun 2019. arXiv: 1906.08172. 5

[23] Dmitry Nikitko. StyleGAN – encoder for official tensorflow implementation. https://github.com/puzer/stylegan-encoder/, 2019. 2

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and et al. PyTorch: An imperative style, high-performance deep learning library. *arXiv:1912.01703 [cs, stat]*, Dec 2019. arXiv: 1912.01703. 5

[25] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: Text-driven manipulation of StyleGAN imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2085–2094, October 2021. 1, 2

[26] Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *Proc. CVPR*, 2020. 2

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and et al. Learning transferable visual models from natural language supervision. In *Proc. ICML*, Feb 2021. arXiv: 2103.00020. 2

[28] Yipeng Qin Rameen Abdal and Peter Wonka. Image2StyleGAN: How to embed images into the StyleGAN latent space? In *Proc. ICCV*, pages 4431–4440, 2019. 2

[29] Yipeng Qin Rameen Abdal and Peter Wonka. Image2StyleGAN++: How to edit the embedded images? In *Proc. CVPR*, pages 8293–8302, Aug 2020. 1, 2

[30] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a StyleGAN encoder for image-to-image translation. In *Proc. CVPR*, June 2021. 2

[31] Daniel Roich, Ron Mokady, Amit H. Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *arXiv:2106.05744 [cs]*, Jun 2021. arXiv: 2106.05744. 2, 6

[32] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of GANs for semantic face editing. In *CVPR*, 2020. 2

[33] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. InterFaceGAN: Interpreting the disentangled face representation learned by GANs. *TPAMI*, 2020. 1, 2, 6, 7, 8

[34] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in GANs. In *Proc. CVPR*, 2021. 1, 2

[35] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zöllhofer, and Christian Theobalt. StyleRig: Rigging StyleGAN for 3d control over portrait images. In *Proc. CVPR)*. IEEE, June 2020. 1, 2

[36] Carlos Eduardo Thomaz and Gilson Antonio Giraldi. A new ranking method for principal components analysis and its application to face image analysis. *Image and Vision Computing*, 28(6):902–913, 2010. 5, 7

[37] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for StyleGAN image manipulation. *arXiv:2102.02766 [cs]*, Feb 2021. arXiv: 2102.02766. 2, 3, 5, 8

[38] M. A. O. Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, ECCV '02, page 447–460, Berlin, Heidelberg, 2002. Springer-Verlag. 2

[39] M. A. O. Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: TensorFaces. In *Proc. ECCV*, page 447–460, Berlin, Heidelberg, 2002. Springer-Verlag. 2

[40] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. Face transfer with multilinear models. In *Proc. ACM SIGGRAPH*, pages 426–433, 2005. 2

[41] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for StyleGAN image generation. In *Proc. CVPR*, Dec 2020. 1, 2

[42] Chao Yang and Ser-Nam Lim. Unconstrained facial expression transfer using style-based generator, 2019. 2

[43] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and M.J. Rosato. A 3D facial expression database for facial behavior research. In *Proc. FG2006*, pages 211–216, 2006. 5

[44] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc CVPR*, 2018. 2, 5, 8

[45] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain GAN inversion for real image editing. In *Proc. ECCV*, 2020. 2

[46] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. *Proc. ECCV 2016)*, Dec 2018. arXiv: 1609.03552. 2

A.3 Paper III:
Controllable GAN Synthesis Using Non-Rigid Structure-from-Motion

Controllable GAN Synthesis Using Non-Rigid Structure-from-Motion

René Haas Stella Graßhof Sami S. Brandt
IT University of Copenhagen, Denmark
{renha,stgr,sambr}@itu.dk

Abstract

In this paper, we present an approach for combining non-rigid structure-from-motion (NRSfM) with deep generative models, and propose an efficient framework for discovering trajectories in the latent space of 2D GANs corresponding to changes in 3D geometry. Our approach uses recent advances in NRSfM and enables editing of the camera and non-rigid shape information associated with the latent codes without needing to retrain the generator. This formulation provides an implicit dense 3D reconstruction as it enables the image synthesis of novel shapes from arbitrary view angles and non-rigid structure. The method is built upon a sparse backbone, where a neural regressor is first trained to regress parameters describing the cameras and sparse non-rigid structure directly from the latent codes. The latent trajectories associated with changes in the camera and structure parameters are then identified by estimating the local inverse of the regressor in the neighborhood of a given latent code. The experiments show that our approach provides a versatile, systematic way to model, analyze, and edit the geometry and non-rigid structures of faces.

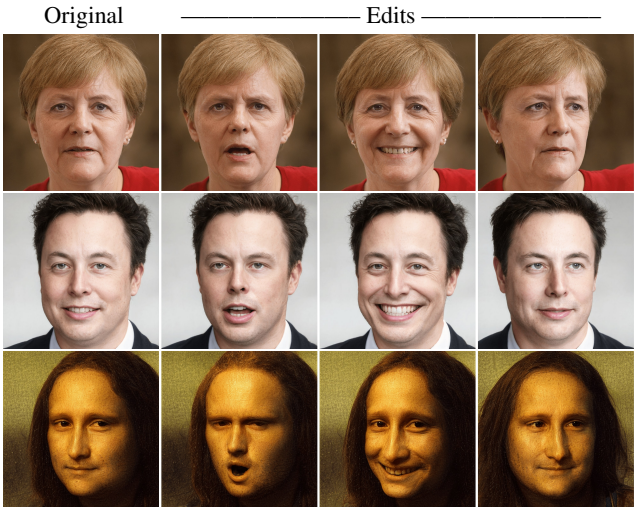


Figure 1. **Semantic editing of real image.** Our method parameterizes the latent space of StyleGAN in terms of camera and shape parameters. This allows for editing of rotation, translation, and non-rigid shape deformation of the synthesized images. Coupled with a strong latent encoder, like e4e [45] or HyperStyle [5], our method allows for semantic editing of real images. Here we show two non-rigid changes corresponding to facial expressions (2nd and 3rd column) as well as a rigid edit corresponding to camera orientation (4th column).

1. Introduction

In recent years, Generative Adversarial Networks (GANs) [15] have seen rapid improvements in image quality as well as training stability. GANs have achieved remarkable results in tasks such as image synthesis [23–27], image-to-image translation [11, 12, 36], semantic editing [1,2,21,33,39,43,47] as well as regression tasks [32]. Especially the StyleGAN [24–27] family of models show state-of-the-art results in unconditional synthesis human faces images. However, the standard StyleGAN architecture provides no way to directly control semantics like the pose and expression of the generated images. This has led to a large interest in finding semantic directions in the latent space of StyleGAN which controls specific semantic attributes such as pose, expression, hairstyle, illumination, etc.

The non-rigid structure-from-motion (NRSfM) problem is a difficult, under-constrained problem with a long history in computer vision. NRSfM aims at obtaining the three-dimensional reconstruction of a scene with dynamical deformable structures from a sequence of 2D correspondences. Given a set of 2D correspondences, the standard assumption is that the deformable 3D shape is a linear combination of basis shapes; the camera information, describing how the 3D structure is projected onto the image plane, also needs to be recovered. In this work, we incorporate a sparse 3D model based on NRSfM into a generative model like StyleGAN. This is interesting for two reasons: first, this allows us to find trajectories in the latent space corresponding to well-defined semantic attributes corresponding to the camera geometry and non-rigid structure. Second, using a generative model in conjunction with NRSfM provides a

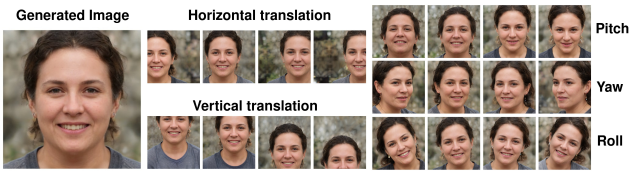


Figure 2. **Rigid edits to rotation and translation.** Our method discovers trajectories in latent space corresponding to arbitrary rotations and translation.

way to obtain an *implicit* dense 3D reconstruction by using only the sparse 2D inputs. By this, we refer to the fact that we are able to view the dense 2D face from an arbitrary 3D orientation, as if we had an explicit dense 3D reconstruction available. In other words, our approach enables dense image synthesis of novel shapes from arbitrary view angles and non-rigid deformation without the need for an explicit dense 3D reconstruction.

In Fig. 1, we demonstrate semantic editing of real images by using our method in conjunction with a recent method for GAN inversion [45]. In Fig. 2 we show latent trajectories corresponding to changes in the rigid camera parameters such as rotation and translations. Note that such edits are only possible if the generator has been trained on a data set that contains such variations, *i.e.*, of translation and roll rotation. In other words, we need an unaligned data set, like FFHQU [25].

Our method utilizes a sparse backbone that is a 3D model based on the approach for NRSfM given in [8, 16]. The 3D model is constructed using solely 2D landmarks extracted from synthetic face images generated by StyleGAN, thus our approach requires no 3D supervision.

In this approach, we first factorize the measurement matrix, consisting of corresponding 2D landmark points, into a rigid and non-rigid part each composed of camera and 3D shape information respectively. Any arbitrary 3D shape can then be represented as the sum of a rigid basis shape and a linear combination of rank-one non-rigid basis shapes. Our approach provides a way to recover a set of expansion coefficients that contains all the information about the 3D reconstruction of the extracted 2D face landmarks. Additionally, for each set of 2D landmarks, we recover a projection matrix, describing the camera information for projecting the 3D shapes onto the image plane as well as information about the orientation of the recovered 3D structure.

We then proceed to connect the information recovered from the sparse 2D landmarks to the latent space of StyleGAN by training a regressor in the form of a multilayer perceptron (MLP) network to regress the shape and camera information directly from the latent codes. By estimating the local inverse of the regressor at a given latent code, we can identify trajectories in latent space corresponding to changes in camera or non-rigid geometry, while preserving other attributes of the generated image, like identity, texture, and illumination. We show that the regressor network can be used for semantic editing of latent codes, either by using the first-order Taylor expansion of the trained network to define linear directions in latent space or by using the prediction of the network as a loss term for a gradient-based optimization algorithm.

As noted in [46], performing semantic editing in StyleGAN using only 2D landmarks is a very challenging problem since the 2D coordinates are extremely localized compared to more global attributes like age or gender.

In summary, we propose an editing framework that relies solely on sparse 2D landmarks. From the landmarks, we use NRSfM to extract camera and shape parameters describing the underlying 3D geometry. We train a regressor to predict these parameters directly from the latent codes and show how the regressor naturally enables editing of the camera and non-rigid geometry of the generated images.

The main contributions of this paper are the following.

- We propose a framework that incorporates the NRSfM problem into the latent space of generative models.
- Based on NRSfM we suggest a framework to get artistic control over images synthesized by StyleGAN.
- We show how our approach can model the camera, pose, and non-rigid structure of the synthesized images, without an explicit dense 3D reconstruction.
- We propose a general method for enabling 3D awareness in 2D GANs without requiring any retraining or changes to the generator architecture.
- We propose a regularization technique that preserves the identity of the synthesized faces during the edits.

2. Related Work

StyleGAN. The StyleGAN [24–27] generator is inspired by the style transfer literature [14,20] and consists of a *mapping network* f which maps a latent vector $\mathbf{z} \in \mathcal{Z}$, sampled from the standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$ in order to obtain an intermediate representation $\mathbf{w} \in \mathcal{W}$. The latent space $\mathcal{W}$ is more disentangled than $\mathcal{Z}$ [26]. To synthesize an image, the latent code $\mathbf{w}$ is copied and fed to each synthesis block of the *synthesis network* G which produces the final image. Instead of feeding the same vector to each of the synthesis blocks, if the vectors are allowed to differ, the resulting space is typically denoted as $\mathcal{W}+$. It has been shown that using $\mathcal{W}+$ space can lead to lower reconstruction loss when performing GAN inversion [35, 50], however at the cost of lower editability [45] of the resultant latent codes.

Semantic Editing. Several methods have been proposed to enable semantic edits of the images produced by StyleGAN. InterFaceGAN [38, 39] enables editing of binary semantic attributes like left/right pose, gender, presence or ab-

sence of smile, etc. Here, a set of latent codes are first sampled and the images are annotated using pre-trained binary classifiers. Following the annotation step, a support vector machine was fitted on the labeled data for each binary semantic attribute. The normal vector for the supporting hyperplane then defines the semantic direction in latent space. Another approach for semantic editing is GANSpace [21] which proposes to use PCA on sampled latent codes to find semantic directions in an unsupervised fashion. Another related approach also factorizes the weights of the trained generator [40, 42] rather than the latent codes. Both methods then change the semantics of the generated images by perturbing latent codes in the direction of the found semantic directions. Additionally, [2] uses normalizing flows for attribute-conditioned semantic editing and explores both linear and non-linear trajectories in latent space. Another related approach, StyleRIG [43] proposes semantic editing in StyleGAN using 3D morphable models [7]. Recently it was proposed to regard the space of channel-wise style parameters after the learned affine transformation in each block in the StyleGAN synthesis network as a separate latent space, complementing the previously mentioned $\mathcal{Z}$, $\mathcal{W}$ and $\mathcal{W}+$ spaces. This latent space was named StyleSpace and denoted as $\mathcal{S}$ [47]. It has been shown that $\mathcal{S}$ space has superior disentanglement properties, especially in StyleGAN3 [4, 25], compared to $\mathcal{W}$ space thus enabling fine-grained and highly localized edits, like the closing of the eyes or changes to hair color [47].

Inversion. For purposes involving the editing of real images, it is necessary to find a good latent representation. That is, we need to find a latent code that, when passed to the generator, reconstructs the target image. This problem is known as GAN inversion. Techniques for GAN inversion have either used optimization-based approaches, where the latent code is directly optimized in order to reconstruct the target image [1, 27, 35] or encoder-based approaches, where a target image is directly mapped into the latent space [3, 34, 36] or hybrid approaches [6, 50].

Recent work [45] suggests that there is a trade-off between distortion and editability when selecting which latent space to project a given target image into. Projecting images into the extended $\mathcal{W}+$ space typically leads to higher reconstruction quality [35], *i.e.*, produces a generated image which is more similar to the target image. However, latent codes in $\mathcal{W}+$ are generally less suitable for semantic editing than latent codes in the native $\mathcal{W}$ space.

The e4e encoder proposed in [45] seeks to find a good trade-off between reconstruction and editability by projecting images into $\mathcal{W}+$ but constraining the latent codes to be close to $\mathcal{W}$. Recently [37] shows that real images can be embedded into $\mathcal{W}$ space by fine-tuning the trained generator around the target image, thus circumventing the need for projecting into $\mathcal{W}+$ space. In [3], a combination of the iterative and encoder-based methods is proposed. Here the encoder predicts the residual with respect to the current estimate of the latent code and thus is able to refine the latent code using only a few forward passes of the encoder in a process referred to as iterative refinement. Recently, [5] proposed to unite the ideas of fine-tuning the generator from [37] with the iterative refinement from [3] by introducing a hypernetwork which predicts how the parameters of the generator should be changed in order to faithfully embed a given real image into the native, and more editable, $\mathcal{W}$ space.

Explicitly 3D aware GANs. Several works have investigated incorporating explicit 3D understanding into GANs [17, 31, 48]. Compared to these, our approach can be used to control the 3D structure in existing 2D GANs without the need for adaptation of the generator architecture nor does our approach require any retraining.

NRSfM. Structure-from-motion (SfM) deals with the problem of inferring the scene geometry and camera information from image sequences. In [44], an orthographic camera model was assumed to infer rigid shape and motion by a factorization of the measurement matrix. In [10], this problem was formulated to include non-rigid deformations by assuming that a shape is a linear combination of 3D basis shapes, hence proposing an approach for non-rigid structure-from-motion (NRSfM). Various works have followed up on this approach over the years this is still an area of active research [22].

Recently, there have been attempts to solve the NRSfM problem by employing neural networks. However, most require a large training data set [29], 3D supervision, or an assumption of an orthographic camera model [29, 41]. Specifically, [29] formulates the NRSfM problem as a multi-layer block sparse dictionary learning problem converted into a deep neural network. In neural NRSfM [41], the authors rely on dense 2D point tracks to recover dense 3D representations, and train an auto-decoder-based model with subspace constraints in the Fourier domain. Our method differs from these works in several aspects, because (1) it relies only on sparse 2D points, (2) it does not rely on a block structure, and (3) it assumes an affine camera model. This makes our approach direct, lightweight, fast, and efficient.

3. Method

Let $I = G(\mathbf{w})$ be an image generated by the StyleGAN generator by the latent code $\mathbf{w}$. Our goal is to locally parameterize the manifold of latent codes, in the neighborhood of a fixed latent code $\mathbf{w}_0$, by an *attribute vector* $\mathbf{q}$ so that

$$\mathbf{w} = \Omega_{\mathbf{w}_0}(\mathbf{q}), \tag{1}$$

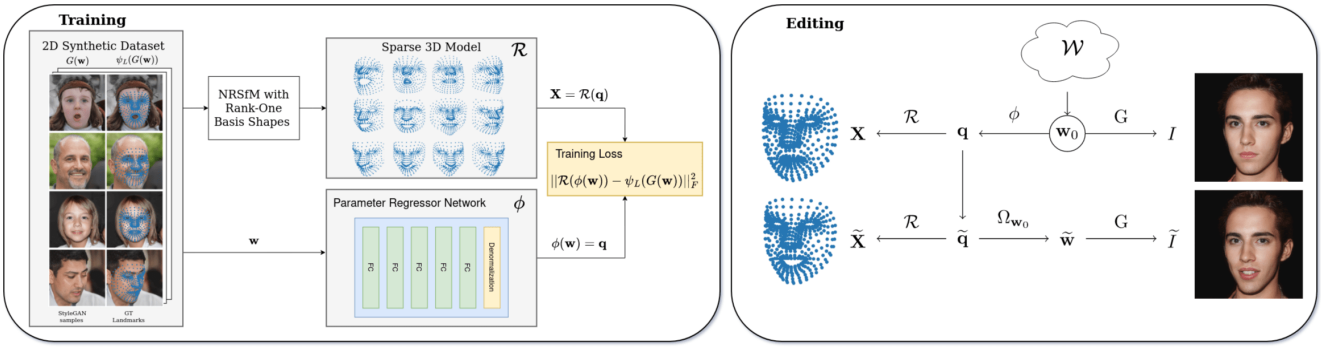


Figure 3. **Overview of our method.** We first create a sparse 3D model $\mathcal{R}$ of facial landmarks from a data set of 2D landmarks $\mathbf{X}$ using NRSfM. The 3D model is parameterized by an attribute vector $\mathbf{q}$ which contains information about the camera, rotation, and non-rigid 3D structure. We then train a regressor ϕ to predict the parameters $\mathbf{q}$ directly from latent codes $\mathbf{w}$. Once the regressor is trained, it can be used for semantic editing. Given a latent code $\mathbf{w}_0$ with corresponding attribute vector $\mathbf{q}_0$ we can define a different, target attribute vector $\widetilde{\mathbf{q}}$ and transfer it onto $\mathbf{w}_0$ using the transformation $\Omega_{\mathbf{w}_0}$ which depends on the regressor ϕ.

where $\mathbf{q}$ describes the *pose*, *shape*, and *camera* information of the generated image. This formulation facilitates the transfer of the target attributes $\mathbf{q}$ onto the latent code $\mathbf{w}_0$ to obtain an edited code $\mathbf{w}$ where only the target attributes have changed in the image, while preserving all other attributes such as identity, texture, and illumination.

Our method is composed of three distinct elements. (1) The *sparse back-bone* relies on a pre-trained landmark extractor ψ_L, which extracts the 2D landmarks $\mathbf{X} = \psi_L(I)$, from a generated image I coupled with a closed-form parameterization for the 2D landmarks as $\mathbf{X} = \mathcal{R}(\mathbf{q})$, where $\mathcal{R}$ maps the 3D shape defined by the attribute vector $\mathbf{q}$ onto the image plane. (2) The *attribute regressor* ϕ predicts the attribute vector $\mathbf{q} = \phi(\mathbf{w})$ from the latent code $\mathbf{w}$, where the regressor is trained by minimizing the squared distance between the ground truth landmarks $\mathbf{X} = (\phi_L \circ G)(\mathbf{w})$ and predicted landmarks $\widehat{\mathbf{X}} = (\mathcal{R} \circ \phi)(\mathbf{w})$. (3) The *regression inversion* constructs the local inverse of the regressor ϕ around the latent code $\mathbf{w}_0$, *i.e.*, finds the local parameterization of the latent space so that $\mathbf{w} = \Omega_{\mathbf{w}_0}(\mathbf{q})$, where $\phi(\mathbf{w}_0) = \mathbf{q}_0$. In Fig. 3 we provide a graphical overview of our approach.

The remaining part of this section is organized as follows. In Section 3.1 we introduce the landmark parameterization $\mathcal{R}(\mathbf{q})$ and detail how the 3D basis shapes can be recovered from a data set of sparse 2D landmarks. The training of the attribute network ϕ is discussed in Section 3.2 and finally, in Section 3.3 we show how the regressor ϕ is used to facilitate highly interpretable semantic editing.

3.1. Rank-one model

The rank-one approach for non-rigid structure-from-motion, proposed in [8, 9, 16], is an affine camera model for non-rigid structure-from-motion which is able to recover 3D structure from sparse 2D correspondences using rank-one basis shapes. In this paper, we frame the model as a parameterization of the space of possible 2D shapes in terms of camera, rotation, translation, and shape parameters. We propose to write the model of [8, 9] in closed-form as

$$\mathcal{R}(\mathbf{q}) = \underbrace{\mathbf{K}[\mathbf{I}_2|\mathbf{0}]\mathbf{R}(\boldsymbol{\theta})}_{\mathbf{M}} \left[\mathbf{B}_0 + \sum_{k=1}^{K} \alpha_k \mathbf{B}_k\right] + \mathbf{t} \otimes \mathbf{1}_L^{\mathrm{T}}, \quad (2)$$

where $\mathbf{K} \in \mathbb{R}^{2\times 2}$ an upper triangular matrix, containing the camera parameters $\mathbf{k} = (k_{11}, k_{12}, k_{22})$. The rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times 3}$ is parameterized in terms of the Euler angles $\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_z)$. The rigid basis shape $\mathbf{B}_0$ describes the average 3D reconstruction while the non-rigid basis shapes $\mathbf{B}_k$ for $k > 0$ describe the non-rigid variation from the rigid basis shape. The expansion coefficients $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_K)$ determine the strength of the contribution of each of the non-rigid basis shapes $\mathbf{B}_k$. Finally, the translation vector $\mathbf{t}$ determines the offset from the origin. In (2), $\otimes$ denotes the Kronecker product, $\mathbf{1}_L \in \mathbb{R}^L$ is a vector of ones, thus $\mathbf{t} \otimes \mathbf{1}_L^{\mathrm{T}} \in \mathbb{R}^{2\times L}$ yields a matrix where $\mathbf{t} \in \mathbb{R}^2$ is repeated L-times column-wise. To summarize, with (2) any 2D shape $\mathbf{X}$ can be parameterized in terms of an attribute vector $\mathbf{q}$ as $\mathbf{X} = \mathcal{R}(\mathbf{q})$ where the attribute vector contains the camera, rotation, shape, and translation parameters as $\mathbf{q} = (\mathbf{k}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{t})$.

In the next section, we see how the rigid basis shape $\mathbf{B}_0$ and non-rigid basis shapes $\mathbf{B}_k$, $k > 0$, can be recovered given a data set of corresponding 2D landmark points.

3.1.1 Non-rigid Factorization.

Given N 2D shapes $\mathbf{X}_n \in \mathbb{R}^{2\times L}$, we stack them into a measurement matrix $\mathcal{X} \in \mathbb{R}^{2N\times L}$. Our aim is to factorize $\mathcal{X}$ into a rigid $\mathbf{X}_0$ and non-rigid $\delta\mathbf{X}$ part such that

$$\mathcal{X} = \mathcal{X}_0 + \delta\mathcal{X} = \mathbf{M}_0\mathbf{B}_0 + \delta\mathbf{M}\delta\mathbf{B}. \quad (3)$$

To recover the rigid basis shape $\mathbf{B}_0$ from (2) we first calculate the singular value decomposition (SVD) of the measurement matrix as $\mathcal{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}$. The rigid part $\mathcal{X}_0$ is then constructed by selecting the three dominant singular vectors such that

$$\mathcal{X}_0 = \mathbf{U}_0\mathbf{\Lambda}_0\mathbf{V}_0^{\mathrm{T}} = \mathbf{M}_0\mathbf{B}_0 \quad \text{with} \tag{4}$$

$$\mathbf{M}_0 = \mathbf{U}_0\mathbf{\Lambda}_0 \in \mathbb{R}^{2N\times 3}, \quad \mathbf{B}_0 = \mathbf{V}_0^{\mathrm{T}} \in \mathbb{R}^{3\times L}. \tag{5}$$

The matrix $\mathbf{M}_0$ contains the N affine projection matrices $\mathbf{M}_n$, associated with each shape in the data set, which are stacked on top of each other in $\mathbf{M}_0$.

To recover the non-rigid basis shapes $\mathbf{B}_k$, we subtract the rigid part from the measurement matrix, *i.e.*, $\delta\mathcal{X} = \mathcal{X} - \mathcal{X}_0$, and calculate the SVD of the remaining part as

$$\delta\mathcal{X} = \delta\mathbf{U}\delta\mathbf{\Lambda}\delta\mathbf{V}^T = \delta\mathbf{M}\delta\mathbf{B}. \tag{6}$$

In the following, we use $\delta\mathbf{B} = \delta\mathbf{V}^{\mathrm{T}} \in \mathbb{R}^{L\times L}$ to construct the non-rigid basis shapes as $\mathbf{B}_k = \mathbf{d}_k\mathbf{b}_k^{\mathrm{T}}$, where $\mathbf{b}_k^{\mathrm{T}}$ is the kth row of $\delta\mathbf{B}$, and $\mathbf{d}_k$ is a 3×1 unit vector which will be determined by gradient-based optimization. Now our goal is to recover $\mathbf{D} = [\mathbf{d}_1, \cdots, \mathbf{d}_K] \in \mathbb{R}^{3\times K}$ which defines the non-rigid basis shapes. In [8, 9, 16], $\mathbf{D}$ was recovered by an alternating least squares optimization scheme by exploiting the orthonormality of the non-rigid basis shapes. Here we use gradient-based optimization instead. For this purpose, it is convenient to write the factorization of the measurement matrix $\mathcal{X}$ as

$$\mathcal{X} = \mathbf{M}_0\mathbf{B}_0 + \mathbf{M}^{\alpha}\mathbf{B}, \tag{7}$$

where

$$\begin{aligned}\mathbf{M}^{\alpha} &= (\boldsymbol{\alpha}\otimes\mathbf{1}_{2\times 3})\odot(\mathbf{1}_K\otimes\mathbf{M}_0)\\ &= \begin{bmatrix}\alpha_{11}\mathbf{M}_1 & \alpha_{12}\mathbf{M}_1 & \cdots & \alpha_{1K}\mathbf{M}_1\\ \alpha_{21}\mathbf{M}_2 & \alpha_{22}\mathbf{M}_2 & \cdots & \alpha_{2K}\mathbf{M}_2\\ \vdots & \vdots & \ddots & \vdots\\ \alpha_{N1}\mathbf{M}_N & \alpha_{N2}\mathbf{M}_N & \cdots & \alpha_{NK}\mathbf{M}_N\end{bmatrix},\end{aligned} \tag{8}$$

where $\odot$ is the Hadamard product and

$$\mathbf{B} = \mathrm{diag}(\mathrm{vec}(\mathbf{D}))(\mathbf{I}_K\otimes\mathbf{1}_3)\delta\mathbf{B} = \begin{bmatrix}\mathbf{d}_1\mathbf{b}_1^{\mathrm{T}}\\ \mathbf{d}_2\mathbf{b}_2^{\mathrm{T}}\\ \vdots\\ \mathbf{d}_K\mathbf{b}_K^{\mathrm{T}}\end{bmatrix} = \begin{bmatrix}\mathbf{B}_1\\ \mathbf{B}_2\\ \vdots\\ \mathbf{B}_K\end{bmatrix}. \tag{9}$$

Then we can jointly find $\mathbf{D}$ and $\boldsymbol{\alpha}$ by minimizing

$$\min_{\mathbf{D},\boldsymbol{\alpha}} ||\widehat{\mathcal{X}}(\mathbf{D},\boldsymbol{\alpha}) - \mathcal{X}||_F^2 + \lambda\sum_{k=1}^{K}(\mathbf{d}_k^{\mathrm{T}}\mathbf{d}_k - 1)^2,\ \lambda\in\mathbb{R}^+, \tag{10}$$

by gradient descent. Once we have found the $\mathbf{D}$ and $\boldsymbol{\alpha}$ which minimizes (10), the non-rigid basis shapes can be constructed using (9). The found basis shapes $\mathbf{B}_i$ completely specify the parameterization in (2).

The parameterization of a new unseen set of landmarks $\mathbf{X}_{\text{new}}$ can be obtained as

$$\mathbf{q}^* = \arg\min_{\mathbf{q}} ||\mathcal{R}(\mathbf{q}) - \mathbf{X}_{\text{new}}||_F^2. \tag{11}$$

3.2. Connection to the latent space

Having found the parameterization $\mathcal{R}$ in (2), we train a MLP network ϕ to regress the parameters $\mathbf{q}$ directly from the latent codes $\mathbf{w}$ such that $\phi(\mathbf{w}) = \widehat{\mathbf{q}}$. Predicting $\mathbf{q}$ is equivalent to predicting the landmarks of the generated images as $\mathcal{R}(\phi(\mathbf{w})) = \widehat{\mathbf{X}}$. We train the network ϕ to minimize the objective function

$$\mathcal{L}(\mathbf{w}) = \|\mathcal{R}(\phi(\mathbf{w})) - \psi_L(G(\mathbf{w}))\|_F^2, \tag{12}$$

where ψ_L is some pre-trained landmark extractor.

3.3. Semantic Editing

In the following, we provide an analytic as well as a gradient-based approach for locally inverting the trained network ϕ, to control the pose and non-rigid shape of images generated by StyleGAN. For the *analytic approach*, the first order Taylor expansion of ϕ around $\mathbf{w}_0$ yields

$$\phi(\mathbf{w}) = \phi(\mathbf{w}_0) + \mathbf{J}|_{\mathbf{w}=\mathbf{w}_0}(\mathbf{w} - \mathbf{w}_0), \tag{13}$$

where $\mathbf{J}|_{\mathbf{w}=\mathbf{w}_0}$ is the Jacobian of ϕ evaluated at $\mathbf{w}_0$. Now since $\phi(\mathbf{w}_0) = \mathbf{q}_0$ we can rewrite this as

$$\mathbf{w} = \mathbf{w}_0 + \mathbf{J}^{\dagger}(\mathbf{q} - \mathbf{q}_0), \tag{14}$$

where $\mathbf{J}^{\dagger}$ is the Moore-Penrose pseudo-inverse of $\mathbf{J}|_{\mathbf{w}=\mathbf{w}_0}$. This allows us to edit a latent code $\mathbf{w}_0$ with associated 2D landmarks $\mathbf{X}_0$ parameterized by $\mathbf{q}_0$ as $\mathbf{X}_0 = \mathcal{R}(\mathbf{q}_0)$ in such a way as to obtain a new latent code $\mathbf{w}$ with a corresponding set of landmarks parameterized by $\mathbf{q}$.

The analytic method described in (14) requires evaluating $\mathbf{J}$ at $\mathbf{w}_0$ and defines a linear path in latent space. As an alternative to (14) we propose a *gradient-based approach* where we directly minimize the difference between the network prediction $\phi(\mathbf{w})$ and a target attribute vector $\mathbf{q}_{\text{target}}$ via

$$\min_{\mathbf{w}} \|\phi(\mathbf{w}) - \mathbf{q}_{\text{target}}\|^2 + \lambda\mathcal{D}(G(\mathbf{w}), G(\mathbf{w}_0)), \tag{15}$$

where $\mathcal{D}(\cdot,\cdot)$ is an image similarity metric such as Learned Perceptual Image Patch Similarity (LPIPS) [49] or Arcface [13], which we employ for regularization purposes. The gradient-based editing is analogous to what is proposed in [46]. However, here we allow for the passing of gradients

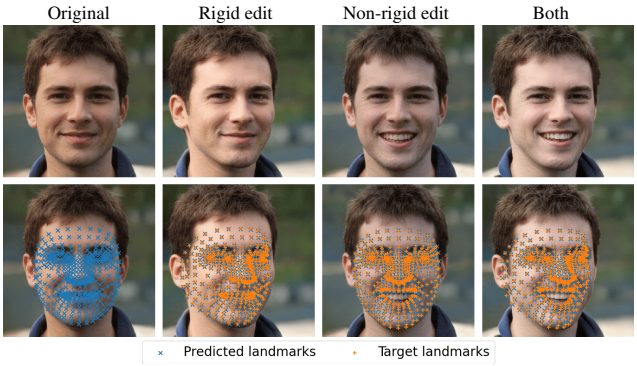


Figure 4. **Rigid and non-rigid edits.** Our approach disentangles rigid edits (rotation) from non-rigid edits (facial expression). We observe that the predicted landmarks agree well with the target landmarks for both types of edits.

through the generator G in order to calculate the identity loss in (15).

In Fig. 4 we visualize the landmarks predicted by our regressor from the latent code as $\mathcal{R}(\phi(\mathbf{w}))$ in blue. Additionally, we showcase semantic editing by changing the latent code $\mathbf{w}$ towards a set of target landmarks $\mathcal{R}(\mathbf{q}_{\text{target}})$ in orange. We show a rigid edit of camera rotation, by changing $\boldsymbol{\theta}$ and a non-rigid edit to facial expression by changing $\boldsymbol{\alpha}$, as well a combination.

4. Experiments

4.1. Implementation Details

We used the StyleGAN2 [27] networks pre-trained on FFHQ [26] as well as StyleGAN3 [25] pre-trained on FFHQU [25]. FFHQ consists of $70K$ face images from flicker and FFHQU is the unaligned version. To construct the model $\mathcal{R}$ in (2) we first sampled $N = 5 \times 10^4$ synthetic images and from each extracted $L = 68$ landmark points with Dlib [28] and $L = 468$ using MediaPipe [30], which were then normalized to the interval $[0, 1]$. In each of the following experiments, we have set the number of non-rigid basis shapes to $K = 12$. Further, we rotated the basis shapes to face the camera when $\boldsymbol{\theta} = \mathbf{0}$ in (2) in order to stabilize the training of the regressor. We trained the regressor, to predict the mean-centered output features $\widehat{\mathbf{q}}$ for each of the N samples. We used the Adam optimizer, 3 hidden layers, each of size 512, and ReLU activation. To evaluate image similarity we use LPIPS and as a metric for identity similarity, we use Arcface [13].

4.2. Model Evaluation

To evaluate our approach we sampled 1000 latent codes $\mathbf{w}$ from the generator G and measured the landmark loss

$$\mathcal{L}_{\text{L}}(\mathbf{w}) = ||(\mathcal{R} \circ \phi)(\mathbf{w}) - (\psi_L \circ G)(\mathbf{w})||^2. \tag{16}$$

Table 1. **Model evaluation.** Comparison of editing results in the latent spaces: $\mathcal{Z}$, $\mathcal{W}$, and $\mathcal{W}+$ of StyleGAN2 and 3. Performance is measured using different metrics, lower is better.

Model / latent space	$\mathcal{L}_L(\mathbf{w})$	$\mathcal{L}_L(\mathbf{w}_{\text{edit}})$	$\mathcal{L}_\phi$	$\mathcal{L}_\mathcal{R}$	$\mathcal{L}_{\text{ID}}$
sg2 / $\mathcal{Z}$	0.037	0.094	0.029	0.123	0.190
sg2 / $\mathcal{W}$	0.006	0.026	0.024	0.057	0.331
sg2 / $\mathcal{W}+$	0.008	0.036	0.058	0.181	0.019
sg3 / $\mathcal{Z}$	0.021	0.036	0.032	0.063	0.264
sg3 / $\mathcal{W}$	0.007	0.019	0.028	0.045	0.296
sg3 / $\mathcal{W}+$	0.009	0.021	0.071	0.160	0.034

We then perform a series of edits $\mathbf{w}_{\text{edit}} = \Omega_{\mathbf{w}}(\mathbf{q}_{\text{edit}})$ using the gradient-based method in (15) with Arcface for identity regularization with $\lambda_{\text{ID}} = 0.01$ For each edit, we measure the landmark loss $\mathcal{L}_{\text{L}}(\mathbf{w}_{\text{edit}})$ as well as three additional losses. First, we measure how well the edits results in the correct change in the prediction of the attribute vector with a metric $\mathcal{L}_\phi$ which we define as

$$\mathcal{L}_\phi = ||\phi(\mathbf{w}_{\text{edit}}) - \mathbf{q}_{\text{edit}})||^2. \tag{17}$$

Secondly, we measure how well the new "ground truth" landmarks of the edited latent code agree with the target landmarks

$$\mathcal{L}_\mathcal{R} = ||\mathcal{R}(\mathbf{q}_{\text{edit}}) - (\psi_L \circ G)(\mathbf{w}_{\text{edit}})||^2. \tag{18}$$

Finally, we measure the identity loss $\mathcal{L}_{\text{ID}}$, between the original and edited images.

For this experiment, we used Dlib as the "ground truth" landmark extractor ψ_L and evaluated the full 1024^2 resolution StyleGAN2 and 3 generators, both trained on the aligned FFHQ data set. We show the results in Table 1. The model was better at predicting landmarks in $\mathcal{W}$ and $\mathcal{W}+$ compared to $\mathcal{Z}$ space when measuring losses $\mathcal{L}_L(\mathbf{w})$ and $\mathcal{L}_L(\mathbf{w}_{\text{edit}})$.

We also observe that the identity loss $\mathcal{L}_{\text{ID}}$ is very low for $\mathcal{W}+$ space, however, $\mathcal{L}_\mathcal{R}$ is also dramatically higher, indicating that it is much harder to change the generated image in such a way that the extracted GT landmarks agree with the specified target when performing edits in $\mathcal{W}+$ space. The same point is supported by the $\mathcal{L}_\phi$ metric with is also substantially higher for $\mathcal{W}+$ space.

4.3. Identity Regularization

We performed a qualitative comparison between the linear (14) and gradient-based method (15), proposed in Section 3.3. Here we edited pose and smile using both methods and show the effect of adding identity regularization, *i.e.*, ArcFace, to the gradient-based method in Fig. 5. In the second column, it can be seen that the linear method is able to define directions in latent space which mostly change the target attribute, *i.e.*, pose or smile, however, we note

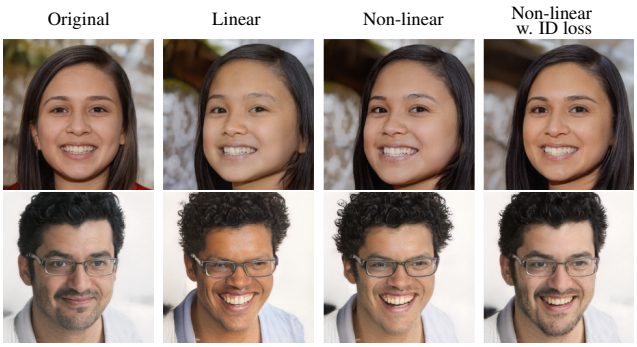


Figure 5. **The effect of identity regularization.** We observe that adding ArcFace to the loss function improves the identity preservation of two edits: rotation (top) and smile (bottom).

that the identity is not preserved well in the edit. This can be alleviated by the gradient-based method which defines a non-linear trajectory in latent space. Further, the gradient-based method in (15) allows for explicit identity regularization using ArcFace which substantially improves the degree of identity preservation for both pose and smile edits as can be seen in column 4 of Fig. 5.

4.4. Attribute Transfer

Our approach enables the transfer of attributes, such as pose or facial expression, from one image to another in a straightforward manner, while preserving other attributes such as identity and illumination. Given two latent codes, $\mathbf{w}_1$ and $\mathbf{w}_2$ with corresponding attribute vectors $\mathbf{q}_1$ and $\mathbf{q}_2$ we can transfer the pose and face shape from $\mathbf{w}_1$ to $\mathbf{w}_2$ by performing the edit $\widetilde{\mathbf{w}}_2 = \Omega_{\mathbf{w}_2}(\mathbf{q}_1)$. Here both $\mathbf{q}_1$ and $\mathbf{q}_2$ can be recovered using either the regressor ϕ or using the minimization procedure in (11). We demonstrate the results of our method in Fig. 6, where we changed the rotation and facial expression of three source images to match different target images, *i.e.*, transferring attributes from the target to the source, while preserving the identity in the source images.

4.5. Rotation and Translation with StyleGAN3

Our method is able to define trajectories in latent space corresponding to roll rotation as well as translations. As noted in [4] roll rotations and translations are a native part of the architecture of the StyleGAN3 generator and can be achieved by manipulating the Fourier features using the four parameters $(\sin\alpha, \cos\alpha, x, y)$ which are obtained from the first learned affine layer of the synthesis network. In comparison, our method can edit rotation and translation directly in the native $\mathcal{W}$ space of StyleGAN3. In Fig. 7, we qualitatively compare the effect of performing roll rotation and translation using our method to the effect of manipulating the Fourier features directly. We note translations look very similar with both methods. However, for roll rotations,

Figure 6. **Attribute transfer.** Our method can edit the rotation and expression of the source image (left column) to match the target image (top row) while preserving identity of the source.

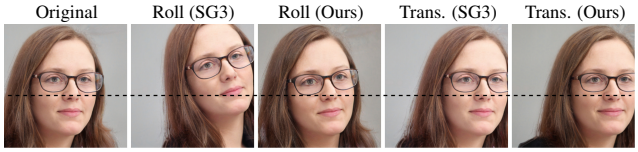
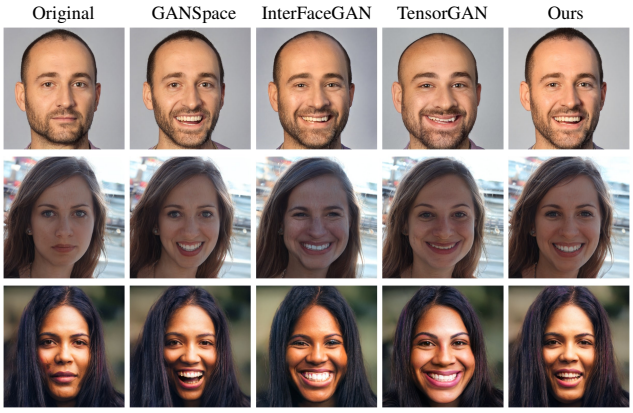


Figure 7. **Comparing our method to Fourier feature editing.** Our method finds a direction for roll rotation where the axis of rotation is at the center of the object. In comparison, manipulating the Fourier features results in an upward movement of the entire face since the axis of rotation is in the middle left border of the image. The vertical dotted line highlights the level of the nose for easier comparison.

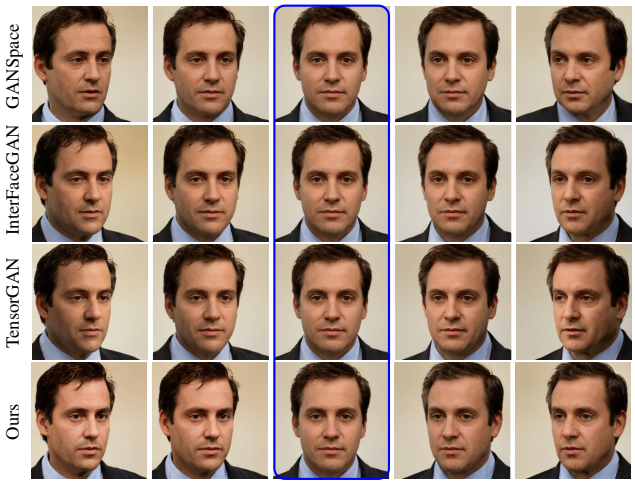
we note that the axis of rotation is located in the middle of the left-hand image border when manipulating the Fourier features (see the location of the nose in Fig. 7), whereas, with our method, the axis of rotation is located at the center of the face.

4.6. Comparison with other Methods

We compared the editing directions corresponding to pose (yaw rotation) and smile with three off-the-shelf techniques for semantic editing: InterFaceGAN [38, 39], GANSpace [21], and TensorGAN [18, 19]. Although our method supports arbitrary 3D rotations in latent space, we focused on editing yaw rotations and smile since previous techniques have also been reported to enable these edits, enabling a direct comparison. A qualitative comparison of the edits to smile and yaw rotations generated by each of the



(a) Smile edits



(b) Yaw rotation edits applied to the original image shown in blue.

Figure 8. **Qualitative comparison to other methods.** We compare editing smile and yaw rotation using our method with equivalent edits using other off-the-shelf techniques.

four methods is shown in Fig. 8a and Fig. 8b respectively.

When evaluating the degree of identity preservation during the semantic edits it can be seen that our method is on par with the competing methods when performing yaw rotations and arguably better when editing smile.

4.7. Editing real images

Coupled with an encoder, our approach facilitates editing of real images. We qualitatively compared the projection and editing results when using our method in conjunction with e4e [45] and HyperStyle [5], respectively. The results are shown in Fig. 9. The two methods operate in different spaces, e4e project images into $\mathcal{W}+$ space while HyperStyle instead makes an initial prediction in $\mathcal{W}$ space and then fine-tunes the generator such that the prediction more faithfully reconstructs the target. Despite the fine-tuning of the generator it is not necessary to retrain the regressor when using HyperStyle for GAN Inversion.

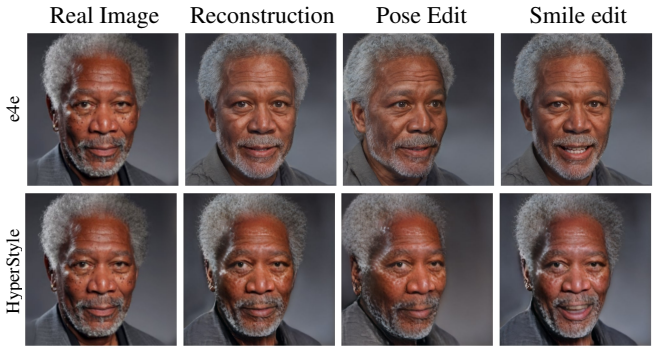


Figure 9. **Editing real images.** Qualitative comparison of projection and editing results when combining our method with two state-of-the-art encoders, e4e [45] and HyperStyle [5] respectively.

5. Conclusions

We presented a framework for highly interpretable image editing in pre-trained 2D GANs. Our framework provides an efficient method to find trajectories in the latent space of GANs which change the generated images according to camera, orientation, and shape parameters. This enables the discovery of trajectories in the latent space corresponding to arbitrary transformations of shape and orientation of the generated images.

In summary, we first used NRSfM to derive a sparse 3D model on the domain of the generator. We then trained a regressor to relate the 3D model to the latent space. We then proposed two methods for using the regressor for semantic editing: a linear method, and a gradient-based method. The latter is similar to the iterative editing algorithm in [46], however, we integrate explicit identity regularization which improves identity preservation.

Our method provides an efficient framework for manipulating the 3D structure of objects generated by 2D GANs. Compared to other methods, our approach is fast compared to existing frameworks for training explicitly 3D aware GANs [17, 31, 48] and compared to [43] our method is lightweight and able to perform rotations and edits to face shape without the need for a 3D morphable model. Since our method only requires access to a landmark extractor trained on the same domain as the generator, our approach does not require any additional training data and can be trained in a fully self-supervised fashion. Further, our approach does not require retraining of the generator or any changes to the generator architecture.

As to limitations, our method allows for adjustments to the position, orientation as well as non-rigid deformation of the face shape of the generated images. Since our method only captures the 3D orientation and face shape our method is not able to add or remove face accessories, eye-glasses, earrings, and hats nor change the skin tone or hair color. Overcoming those limitations is an avenue for future work.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 8293–8302, Seattle, WA, USA, Jun 2020. IEEE. 1, 3

[2] Rameen Abdal, Peihao Zhu, Niloy J. Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Trans. Graph.*, 40(3), May 2021. 1, 3

[3] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021. 3

[4] Yuval Alaluf, Or Patashnik, Zongze Wu, Asif Zamir, Eli Shechtman, Dani Lischinski, and Daniel Cohen-Or. Third time's the charm? image and video editing with stylegan3. *Advances in Image Manipulation Workshop - ECCV 2022*, Jan 2022. 3, 7

[5] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit H. Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proc. CVPR*, 2022. 1, 3, 8

[6] Peter Baylies. Stylegan encoder - converts real images to latent space. https://github.com/pbaylies/styleganencoder/, 2019. 3

[7] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH*, pages 187–194, 1999. 3

[8] Sami S. Brandt and Hanno Ackermann. Non-rigid structure-from-motion by rank-one basis shapes, Apr 2019. arXiv: 1904.13271. 2, 4, 5

[9] Sami Sebastian Brandt, Hanno Ackermann, and Stella Grasshof. Uncalibrated non-rigid factorisation by independent subspace analysis. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 569–578, 2019. 4, 5

[10] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, volume 2, page 690–696. IEEE Comput. Soc, 2000. 3

[11] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1

[12] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1

[13] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proc. CVPR*, pages 4690–4699, 2019. 5, 6

[14] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015. 2

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, page 2672–2680. Curran Associates, Inc., 2014. 1

[16] Stella Graßhof and Sami Sebastian Brandt. Tensor-based non-rigid structure from motion. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, page 2254–2263. IEEE, Jan 2022. 2, 4, 5

[17] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d aware generator for high-resolution image synthesis. In *International Conference on Learning Representations*, 2022. 3, 8

[18] René Haas, Stella Graßhof, and Sami Sebastian Brandt. Tensor-based subspace factorization for stylegan. In *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, pages 1–8, Los Alamitos, CA, USA, dec 2021. IEEE Computer Society. 7

[19] René Haas, Stella Graßhof, and Sami Sebastian Brandt. Tensor-based emotion editing in the stylegan latent space. *arXiv:2205.06102 [cs]*, May 2022. Accepted for poster presentation at AI4CC @ CVPRW. 7

[20] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *Proc. ICCV*, Jul 2017. 2

[21] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *Proc. NeurIPS*, 2020. 1, 3, 7

[22] Sebastian Hoppe Nesgaard Jensen, Mads Emil Brix Doest, Henrik Aanæs, and Alessio Del Bue. A benchmark and evaluation of non-rigid structure from motion. *International Journal of Computer Vision*, 129(4):882–899, Apr 2021. 3

[23] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *Proc. ICLR*, Feb 2018. 1

[24] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020. 1, 2

[25] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021. 1, 2, 3, 6

[26] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*, pages 4396–4405, 2019. 1, 2, 6

[27] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020. 1, 2, 3, 6

[28] Davis E. King. Dlib-ml: A Machine Learning Toolkit. *J. Mach. Learn. Res.*, 10:1755–1758, Dec. 2009. 6

[29] Chen Kong and Simon Lucey. Deep Non-Rigid Structure From Motion. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1558–1567, Oct. 2019. ISSN: 2380-7504. 3

[30] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, and et al. Mediapipe: A framework for building perception pipelines. *arXiv:1906.08172 [cs]*, Jun 2019. arXiv: 1906.08172. 6

[31] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 11448–11459, Nashville, TN, USA, Jun 2021. IEEE. 3, 8

[32] Yotam Nitzan, Rinon Gal, Ofir Brenner, and Daniel Cohen-Or. Large: Latent-based regression through gan semantics. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19217–19227, 2022. 1

[33] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2085–2094, October 2021. 1

[34] Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, and Jose M. Álvarez. Invertible conditional gans for image editing. In *NIPS 2016 Workshop on Adversarial Training*, Nov 2016. 3

[35] Yipeng Qin Rameen Abdal and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proc. ICCV*, pages 4431–4440, 2019. 2, 3

[36] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 1, 3

[37] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Trans. Graph.*, 2021. 3

[38] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020. 2, 7

[39] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *TPAMI*, 2020. 1, 2, 7

[40] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *CVPR*, 2021. 3

[41] Vikramjit Sidhu, Edgar Tretschk, Vladislav Golyanik, Antonio Agudo, and Christian Theobalt. Neural Dense Non-Rigid Structure from Motion with Latent Space Constraints. In *European Conference on Computer Vision (ECCV)*, 2020. 3

[42] Nurit Spingarn, Ron Banner, and Tomer Michaeli. GAN Steerability without optimization. In *International Conference on Learning Representations*, 2021. 3

[43] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zöllhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images, cvpr 2020. In *Proc. CVPR)*. IEEE, june 2020. 1, 3, 8

[44] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992. 3

[45] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for StyleGAN image manipulation. *ACM Transactions on Graphics*, 40(4):133:1–133:14, July 2021. 1, 2, 3, 8

[46] Hui-Po Wang, Ning Yu, and Mario Fritz. Hijack-gan: Unintended-use of pretrained, black-box gans. In *Proc. CVPR)*, page 10, 2021. 2, 5, 8

[47] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proc. CVPR*, Dec 2020. 1, 3

[48] Doğa Yılmaz, Furkan Kınlı, Barış Özcan, and Furkan Kıraç. [re] lifting 2d styleGAN for 3d-aware face generation. In *ML Reproducibility Challenge 2021 (Fall Edition)*, 2022. 3, 8

[49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc CVPR*, 2018. 5

[50] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *Proc. ECCV*, 2020. 2, 3

A.4 Paper IV: Exploring a Digital Art Collection through Drawing Interactions with a Deep Generative Model

Exploring a Digital Art Collection through Drawing Interactions with a Deep Generative Model

Christian Sivertsen
csiv@itu.dk
IT University of Copenhagen
Copenhagen, Denmark

René Haas
renha@itu.dk
IT University of Copenhagen
Copenhagen, Denmark

Halfdan Hauch Jensen
halj@itu.dk
IT University of Copenhagen
Copenhagen, Denmark

Anders Sundnes Løvlie
asun@itu.dk
IT University of Copenhagen
Copenhagen, Denmark

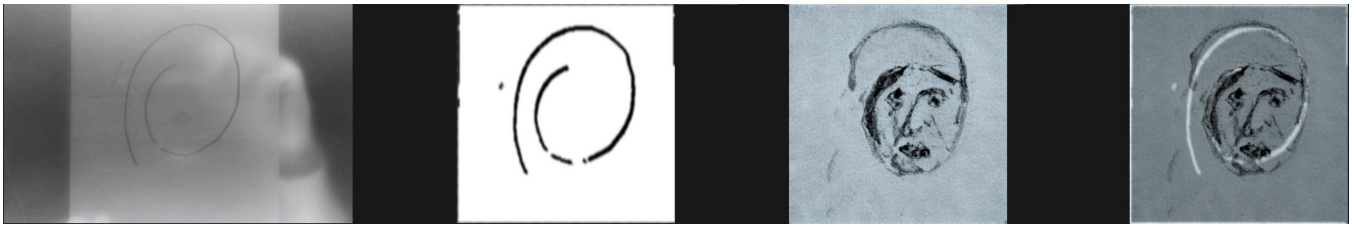


Figure 1: Inferring a Munch-like sketch from a hand-drawn line. From left to right: 1) infrared webcam image from beneath the drawing surface. 2) The cleaned input image. 3) The synthesized sketch. 4) A composite image of the input image over the synthesized sketch.

ABSTRACT

New Snow is an interactive drawing table that investigates human interaction with a deep generative model based on Edvard Munch's sketching practice. Through drawings with pen and paper, the user can interact with the model which will return synthetic sketches based on the input drawings in real time. The model is a reflection of the training data, and it is thus constrained to representing images within the latent space of Edvard Munch's sketching practice. As the user familiarizes themselves with the model it allows them to become sensitized to the visual aesthetic belonging to this practice. This potential for familiarization with the aesthetic of a dataset via the model has implications for human-AI interaction and non-verbal art mediation.

CCS CONCEPTS

• **Human-centered computing** → *Interface design prototyping*; **Interaction design**; **Interaction devices**; • **Applied computing** → **Fine arts**; • **Computing methodologies** → *Machine learning*.

KEYWORDS

sketching, embodied interaction, interaction, stylegan, fine art

CHI EA '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9422-2/23/04.
https://doi.org/10.1145/3544549.3583902

ACM Reference Format:
Christian Sivertsen, René Haas, Halfdan Hauch Jensen, and Anders Sundnes Løvlie. 2023. Exploring a Digital Art Collection through Drawing Interactions with a Deep Generative Model. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems (CHI EA '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3544549.3583902

1 INTRODUCTION

Museums across the world have built up large collections of digitized artwork collections. This vast amount of material can be difficult to present to an audience. Several projects investigate ways to expose museum visitors to large datasets and allow for browsing or reauthoring the content [17–19, 23]. However, collection interfaces are often designed to offer overview and searchability rather than engagement with the artworks. To support the mediation of large art collections, interfaces are necessary that support alternative and embodied modes of engagement.

Tapping into recent developments in image synthesis, New Snow attempts to explore a new way to engage with a large collection of digitized artworks through a novel machine-learning-enabled interface. Deep Generative Models (DGM) are models that can generate images that resemble the training data. These have recently become well-known through the current wave of text-to-image systems like Midjourney, DALL-E [21] and Stable Diffusion [5] that allow for anyone to generate synthetic images in the style of famous artists, that is, artists whose works have a large presence on the internet from where much of the data for the underlying datasets are found. In contrast to these systems, New Snow employs a model trained specifically on the drawings of Edvard Munch (fig. 2), to allow

Figure 2: The upper half of the image shows samples from the original sketch data. The lower half shows synthetic samples from the StyleGAN model

museum visitors to engage with this well-known artist's drawing practice through their own drawing actions. New Snow is an interactive drawing table that offers museum visitors an embodied mode of interaction, where the system responds to the user's drawing by adding lines and patterns generated by the DGM. This simulates an experience of the artist "filling in" the lines drawn by the visitor.

This project aims for three main contributions. First, it enables a way for visitors to engage with a large corpus of artworks that could not feasibly be explored individually, through the proxy of a DGM offering a synthesis of the data. Second, the system allows an embodied and creative engagement through the drawing actions of the visitor, and the interplay between the visitor and the system. Third, the system explores a novel use of a DGM, in which the user's efforts to learn how to interact with the model are offered as a way to learn about the aesthetics of Munch's drawings. As the user investigates the model through the drawing actions, the user learns about its qualities, and by proxy certain qualities of the artworks constituting the underlying dataset. This means that building a mental model of the system becomes a way of learning about the aesthetics of the drawings, and the image synthesis becomes an enabler of the exploration rather than the end goal.

2 RELATED WORK

Large databases of cultural heritage collections are usually accessed through search interfaces letting the users find content based on written prompts and filters. This is useful for situations where the users have a good idea of what they are looking for. However, when the domain is unfamiliar to the user, curation and recommendation are often used as a way of guiding the user to relevant content. Earlier projects have attempted different visualization strategies for large cultural heritage datasets like *T_Visionarium II*[23], *Cloud-browsing*[19] and *ECLOUD WW1*[17] that all utilize large projection surfaces to display content and allow the user to browse through the individual data objects.

The two projects *Draw to Art* [11] and *Draw to Art: Shape Edition* [10] explores visual search by allowing users to draw images on a tablet surface to query a large art database for artworks matching the drawing. In the first version, the match is based on classifying the input image as a word and then returning images relating to that word. In the second version, the system returns images with shapes matching the input image, which is constrained to simple geometric shapes. This difference marks a significant change as it enables exploration that is driven by visual concepts such as composition and shapes.

Human-AI interaction research stipulates that the system should provide the user with clear concepts of its capabilities [2] by being *explainable* or *transparent* e.g. [3, 12]. Another related concept is that of *interpretable* AI [6, 8], asserting that the users of a system should be able to interpret the underlying reasons for the output.

Building expertise in the interaction with image synthesis models is seen in *prompt engineering*, the practice of developing text-based prompts through optimization or exploration that makes text-to-image or text-to-text generation systems generate the content intended by the user. According to Oppenlaender, achieving the best results requires a deep understanding of the underlying dataset [20].

Based on the works above we derive three insights, which have informed the design of New Snow:

(1) We understand the deep generative models as reflecting qualities of the data from which it is trained.
(2) Exploration of datasets and models can happen through non-verbal means.
(3) Learning how to *prompt* a model effectively means building an understanding of the data on which it is trained.

3 CONCEPT AND INTERACTION

New Snow is a project that explores how exploring an image synthesis model using drawing actions as a means of *prompting* lets the user learn about certain qualities of the underlying data.

From the digital collection of MUNCH, we have identified 5800 uncolored, crayon, ink, or pencil drawings made by Edvard Munch (see examples in fig. 2). Based on these drawings we have trained a StyleGAN 2 model [14, 16] and then a pixel2style2pixel (pSp) model [22]. Together this allows us to map drawings made by a user into the model and synthesize sketches from this input.

The prototype consists of a table with a matte transparent surface. The user places a piece of tracing paper on the surface and draws with a pen (fig. 3). Underneath the table, a camera tracks the lines on the paper and sends them to the pixel2style2pixel model. From the lines on the paper, the model synthesizes an image based on Edvard Munch's sketches (fig. 1). This image is projected back onto the tracing paper for the user to see. As the user draws or moves the paper around, the system continuously and multiple times per second updates the synthesized image to match. This

allows the user to explore the qualities of the model through an almost conversational relation to the system.

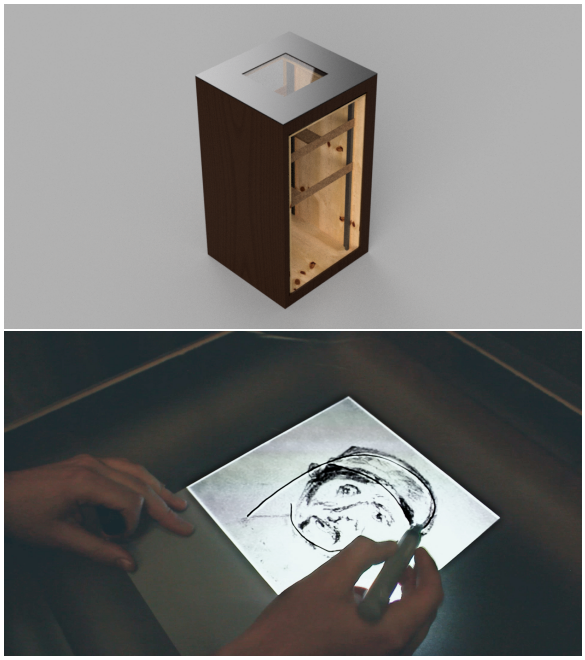


Figure 3: The top image is a render of the prototype with the sides open, so the inner structure can be seen. The image below shows a user interacting with the drawing interface.

The aim of this project is to help the user develop an attention to the aesthetic of Edvard Munch's drawing practice. We are not attempting to explain the technical details of his practice nor the historical or biographical relations. However, through the embodied engagement with the visual aesthetic derived from his drawings, we expect the user to develop a sense of the visual qualities related to Munch's sketching and drawing practice. That does not necessarily require the original works to be reproduced, the aim is rather to create a focus on the dynamics and patterns in Munch's sketching practice that a person not skilled in the act of drawing or analyzing drawings might not otherwise have noticed.

Models like Dall-E, Midjourney, and Stable Diffusion have become famous for their ability to synthesize coherent images from almost any prompt in the style of well-known artists. In comparison, this model provides much more resistance. It does not draw for the user, but it responds to their drawings and attempts to expand and complete them. Due to the nature of the underlying model, the user will have to adapt a particular drawing strategy in order to achieve the greatest level of control, as the model does not respond to symbolic representation but rather the saliency of particular lines constituting an image.

This tension requires the user to explore the workings of the model to understand how it works and what it responds to. In that way, the function of the model is to provide a space to explore rather than being a tool to reach other ends. The exploration is by proxy an exploration of Munch's sketching practice. It asks the user to contemplate and then draw not *what* Munch might have drawn, but *how* he might have drawn.

At the core of the experience is the user interaction with the DGM, however, that interaction happens within an embodied and material context. People's bodily relations to artworks shape how they might cast the art objects in specific cultural roles i.e. as a commodity, a fragile piece of history, or a toy [24]. With this awareness we expect the embodied relation to the drawings in New Snow to influence people's cultural connotations of the drawing activity. Thus we have opted for a physical setup where the user performs the drawing action with an actual pen on paper. First, the paper and pen have other affordances than a touchscreen and pen interface, one being that erasing is not possible, and the drawn image can be moved around on the surface, lifted off the surface, and brought along. Secondly, the tactile feeling of pen and paper differs significantly from the glass surface of a tablet and e-pen and evokes different connotations and importantly a closer material connection to the tools used by Edvard Munch.

4 TECHNICAL DESCRIPTION

The drawing table is built into a flight case 110 cm tall with a semi-transparent polycarbonate window on top. The drawing surface is lit with infrared (IR) light by LEDs within the table to eliminate shadows. An IR-sensitive camera fitted with an 850nm filter records the drawing surface (fig. 4). This is to avoid interference from the visible light cast by the projector. With software made with TouchDesigner, OpenCV, and Python the video feed is pre-processed into binary images, isolating the lines drawn on the paper. This image is submitted to the pixel2style2pixel model and a synthesized drawing is returned within a second. Adjustments are made to saturation and contrast before the projected image interpolates from the current to the new image.

4.1 Machine Learning

DGMs have seen tremendous progress in recent years and Generative Adversarial Networks (GANs) [9] have become one of the most influential deep generative architectures. Recently, inspired by style transfer [7, 13], the StyleGAN family of models [14–16] have been shown to give state-of-the-art results across a wide variety of image generation tasks [4]. Due to the exceptional quality of the images generated by StyleGAN models, the architecture has been called one of the most intriguing and well-studied architectures in recent times [1].

After training, StyleGAN has learned a mathematical space, denoted the *latent* space, where each point in the space corresponds to a unique image. The GAN is trained such that the distribution of the generated images follows the distribution of the images in the training data, both with respect to image quality as well as the internal variation between images in the training data.

The latent space is smooth, which means that if we interpolate between two points in the latent space, e.g two points corresponding to portraits, the corresponding generated images will change gradually from one portrait to the other where each intermediate image is itself fully self-consistent and resembles Munch's style in its own right. Thus the latent space can be seen as a representation of the space between all Munch sketches in the data set.

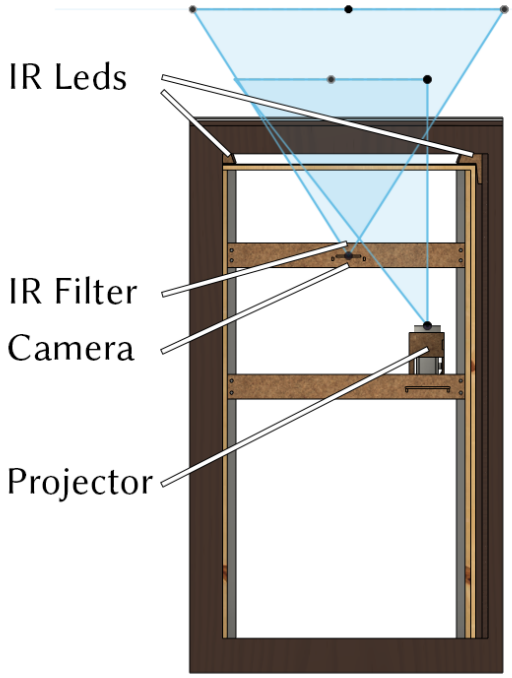


Figure 4: The prototype house a pico projector and a camera that are aligned with the drawing surface. An infrared filter removes the visible light from the camera input. Right below the top are two strips of infrared LED that illuminate the drawing surface. The lower part of the table can house a PC for processing the images.

To allow for direct user interaction with this latent space, we have trained a pSp encoder [22] which is able to map user-provided pen-and-paper sketches, into the latent space, thus transforming the user input to a sketch that follows the style of Munch.

5 AESTHETIC DRAWING STRATEGIES

In preliminary testing of the prototype, we have seen participants engaging actively with the drawing task. Participants apply widely different strategies, and we see indications that certain mental models yield more satisfying interactions than others. When users draw conceptually, e.g. a simplified house or tree, the system generates only limited visual response since these shapes lie far away from the images in the dataset. A more fruitful drawing strategy seems to be drawing one long stroke at first and then looking at the response of the system. Often a variety of more or less defined lines will appear. These lines can be reinforced or challenged by drawing other lines in the same area, which often results in more defined shapes and features, and the process can continue and evolve into a meaningful drawing. These are the strategies that we are interested in exploring and tuning the system to support.

6 CURATING THE DATASET

As the aesthetic qualities in Edvard Munch's sketching practice are mediated through the DGM, particular attention needs to be paid to the ways in which the sketches have been prepared for training and how the chosen model interprets the data. The images constituting the dataset for the StyleGAN model have been cropped from photographs of notebooks or loose paper sheets by human annotators that have made decisions on composition and the tightness of the crop to leave out damaged paper, smudges, handwritten notes, and other artifacts that have been deemed irrelevant for the project. This curation shapes the concept of the images created by the model. It determines what belongs to a drawing, and where on the page certain shapes will appear. Another limitation is the necessity for this type of model to be trained with square images. This requires the input images to be either stretched or cropped to fit this requirement. These issues reappear when the pSp model is trained as the input images are simplifications derived from the syntheses. However, the amount of simplification determines how far from the input image the synthesized images will be visually. This means that a significant part of the interaction design lies in the data curation process, making the iterative loop longer and more time-consuming than when designing heuristic interactive systems.

ACKNOWLEDGMENTS

The authors would like to thank MUNCH, Oslo for helping in developing the concept and funding the prototype. We are thankful to the research division at Random International for the development of the tracking system and AIRLab at IT University of Copenhagen for supporting the physical realization of the prototype.

REFERENCES

[1] Yuval Alaluf, Or Patashnik, Zongze Wu, Asif Zamir, Eli Shechtman, Dani Lischinski, and Daniel Cohen-Or. 2022. Third Time's the Charm? Image and Video Editing with StyleGAN3. http://arxiv.org/abs/2201.13433 arXiv: 2201.13433.

[2] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–13. https://doi.org/10.1145/3290605.3300233

[3] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* 58 (June 2020), 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

[4] Amit H. Bermano, Rinon Gal, Yuval Alaluf, Ron Mokady, Yotam Nitzan, Omer Tov, Or Patashnik, and Daniel Cohen-Or. 2022. State-of-the-Art in the Architecture, Methods and Applications of StyleGAN. https://doi.org/10.48550/ARXIV.2202.14020

[5] Andreas Blattmann, Robin Rombach, Kaan Oktay, and Björn Ommer. 2022. Retrieval-Augmented Diffusion Models. https://doi.org/10.48550/ARXIV.2204.11824

[6] Finale Doshi-Velez and Been Kim. 2017. Towards A Rigorous Science of Interpretable Machine Learning. http://arxiv.org/abs/1702.08608 arXiv:1702.08608 [cs, stat].

[7] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2015. A Neural Algorithm of Artistic Style. http://dblp.uni-trier.de/db/journals/corr/corr1508.html#GatysEB15a

[8] Adarsh Ghosh and Devasenathipathy Kandasamy. 2020. Interpretable Artificial Intelligence: Why and When. *American Journal of Roentgenology* 214, 5 (May 2020), 1137–1138. https://doi.org/10.2214/AJR.19.22145 Publisher: American Roentgen Ray Society.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. , 2672–2680 pages. https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[10] Google Creative Lab, Bastien Girschig, and Romain Cazier. 2020. Draw to Art: Shape Edition. https://experiments.withgoogle.com/draw-to-art-shape

[11] Google Creative Lab, Google Art & Culture Lab, and IYOIYO. 2018. Draw to Art. https://experiments.withgoogle.com/draw-to-art

[12] Joana Hois, Dimitra Theofanou-Fuelbier, and Alischa Janine Junk. 2019. How to Achieve Explainability and Transparency in Human AI Interaction. In *HCI International 2019 - Posters (Communications in Computer and Information Science)*, Constantine Stephanidis (Ed.). Springer International Publishing, Cham, 177–183. https://doi.org/10.1007/978-3-030-23528-4_25

[13] Xun Huang and Serge Belongie. 2017. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. https://doi.org/10.48550/ARXIV.1703.06868

[14] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2020. Training Generative Adversarial Networks with Limited Data.

[15] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-Free Generative Adversarial Networks.

[16] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and Improving the Image Quality of StyleGAN.

[17] Sarah Kenderdine and Heidi McKenzie. 2013. A war torn memory palace: Animating narratives of remembrance. In *2013 Digital Heritage International Congress (DigitalHeritage)*. IEEE, Marseille, France, 315–322. https://doi.org/10.1109/DigitalHeritage.2013.6743755

[18] Sarah Irene Brutton Kenderdine and Tim Hart. 2011. Cultural Data Sculpting: Omni-spatial Visualization for Large Scale Heterogeneous Datasets. In *Museums and the Web 2011: Proceedings*, J. Trant and D. Bearman (Eds.). Archives & Museum Informatics, Toronto. http://conference.archimuse.com/mw2011/papers/cultural_data_sculpting_omnispatial_visualization_large_scale_heterogeneous_datasets

[19] Bernd Lintermann. 2012. Beyond Cinema. Hongik University, Korea. https://www.bernd-lintermann.de/papers/Beyond_Cinema_Lintermann.pdf

[20] Jonas Oppenlaender. 2022. A Taxonomy of Prompt Modifiers for Text-To-Image Generation. http://arxiv.org/abs/2204.13988 arXiv:2204.13988 [cs].

[21] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical Text-Conditional Image Generation with CLIP Latents. https://doi.org/10.48550/ARXIV.2204.06125

[22] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. 2021. Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA, 2287–2296. https://doi.org/10.1109/CVPR46437.2021.00232

[23] Jeffrey Shaw, Neil Brown, Dennis Del Favero, Matt McGinity, and Peter Weibel. 2006. T_Visionarium II. https://www.jeffreyshawcompendium.com/portfolio/t_visionarium-ii/

[24] Christian Sivertsen and Anders Sundnes Løvlie. 2021. Handling digital reproductions of artworks. *Journal of Somaesthetics* 7, 2 (2021), 21.

A.5 Paper V: Discovering Interpretable Directions in the Semantic Latent Space of Diffusion Models

Discovering Interpretable Directions in the Semantic Latent Space of Diffusion Models

René Haas renha@itu.dk
Inbar Huberman-Spiegelglas inbarhub@gmail.com
Rotem Mulayoff rotem.mulayof@gmail.com
Stella Graßhof stgr@itu.dk
Sami S. Brandt sambr@itu.dk
Tomer Michaeli tomer.m@ee.technion.ac.il

Figure 1. **Our semantic image editing.** We present new methods for finding interpretable disentangled semantic directions in the latent space of DDMs. Specifically, we propose a supervised (left) and two unsupervised (right) methods, where the latter finds either global directions based on a collection of images or local directions based on the analysis of a single sample.

Abstract

Denoising Diffusion Models (DDMs) have emerged as a strong competitor to Generative Adversarial Networks (GANs). However, despite their widespread use in image synthesis and editing applications, their latent space is still not as well understood. Recently, a semantic latent space for DDMs, coined 'h-space', was shown to facilitate semantic image editing in a way reminiscent of GANs. The h-space is comprised of the bottleneck activations in the DDM's denoiser across all timesteps of the diffusion process. In this paper, we explore the properties of h-space and propose several novel methods for finding meaningful semantic directions within it. We start by studying unsupervised methods for revealing interpretable semantic directions in pretrained DDMs. Specifically, we show that interpretable directions emerge as the principal components in the latent space. Additionally, we provide a novel method for discovering image-specific semantic directions by spectral analysis of the Jacobian of the denoiser w.r.t. the latent code. Next, we extend the analysis by finding directions in a supervised fashion in unconditional DDMs. We demonstrate how such directions can be found by annotating generated samples with a domain-specific attribute classifier. We further show how to semantically disentangle the found directions by simple linear projection. Our approaches are applicable without requiring any architectural modifications, text-based guidance, CLIP-based optimization, or model fine-tuning.

1. Introduction

Denoising Diffusion Models (DDMs) [37] have emerged as a strong alternative to Generative Adversarial Networks (GANs) [5]. Today, they outperform GANs in unconditional image synthesis [3], a task in which GANs have been dominating in recent years. Besides synthesizing high-quality and diverse images, DDMs can also be used for conditional synthesis tasks by guiding them on various user inputs [10], such as a user-provided reference image [13, 17] or a text-prompt by utilizing Contrastive Language-Image Pretraining (CLIP) [23]. Conditional DDMs have seen great success, particularly in the context of text-based synthesis. Specifically, recent large-scale text-conditional systems like DALL-E [26, 27], Stable Diffusion [28] and Imagen [33] have sparked a surge of research related to text-driven image editing using DDMs [2, 4, 8, 11, 12, 18, 19, 31, 41]. While there has been extensive research on finding disentangled editing directions in the latent space of unconditional GANs [1, 6, 7, 25, 34, 36, 39], comparatively little work has been done on this topic for unconditional DDMs. Despite their

popularity, it is still not well understood how to leverage the latent space of DDMs for semantic image editing in the unconditional setting, *i.e*., in the absence of CLIP-guidance and without conditioning on a reference image.

In this paper, we propose novel editing techniques by utilizing the *semantic latent space* of DDMs which was recently proposed by Kwon *et al*. [14]. The semantic latent space, coined 'h-space', is the space of the deepest feature maps of the denoiser. Our research explores supervised and unsupervised methods for finding semantically interpretable editing directions in unconditional DDMs.

We start by proposing two unsupervised methods. In Sec. 4, we demonstrate that interpretable editing directions, like pose, gender, and age emerge as the principal components in the semantic latent space. Additionally, we propose a novel unsupervised method for discovering image-specific semantic directions resulting in highly localized edits like opening/closing of the mouth and eyes that can also be applied to other samples. We illustrate a selection of these unsupervised editing directions in Fig. 1 (right pane). Next, in Sec. 5, we utilize the linear properties of the semantic latent space and propose a simple supervised method for finding interpretable editing directions, like age and gender or the appearance of glasses or a smile. We illustrate examples of these edits in Fig. 1 (left pane). We demonstrate our approach by annotating samples generated by an unconditional DDM using a pretrained attribute classifier. We further propose a simple method for disentangling directions that affect multiple attributes. Our approaches allow for intuitive and semantically disentangled image editing and can be applied to the latent space of DDMs without requiring any CLIP guidance, fine-tuning, optimization or any adaptations to the architecture of existing DDMs.

2. Related work

2.1. The latent space of diffusion models

GANs have a well-defined latent space suitable for semantic editing. To which extent DDMs possess such a convenient latent space is still a topic of ongoing research. Here we start by reviewing two approaches for defining a latent space in DDMs that facilitate semantic editing.

Using DDIM sampling proposed by Song *et al*. [38], the generative process is a deterministic mapping from a Gaussian noise vector $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to a sampled image $\mathbf{x}_0$. In the DDIM framework, the fully noised image $\mathbf{x}_T$, can be regarded as the latent representation. DDIM has the property that fixing $\mathbf{x}_T$ leads to images with similar high-level features irrespective of the length of the generative process. Furthermore, interpolating between two latent codes $\mathbf{x}_T^{(1)}$ and $\mathbf{x}_T^{(2)}$ leads to images that vary smoothly between the two corresponding endpoint images, $\mathbf{x}_0^{(1)}$ and $\mathbf{x}_0^{(2)}$.

Kwon *et al*. [14] propose h-space for DDMs, the set of bottleneck feature maps of the U-Net [29] across all timesteps, $\{\mathbf{h}_T, \dots, \mathbf{h}_1\}$ as the latent space. Each bottleneck feature map $\mathbf{h}_t$ has a lower spatial dimension but more channels than the output image. They show that semantics can be edited by adding offsets $\Delta\mathbf{h}_t$ to the feature maps during the generative process. To find editing directions, they use an optimization procedure involving CLIP, where the semantics to be edited are described by text prompts. The h-space has the following properties: (i) a direction $\Delta\mathbf{h}_t$ has the same semantic effect on different samples; (ii) the magnitude of $\Delta\mathbf{h}_t$ controls the strength of the edit; (iii) h-space is additive in the sense that applying a linear combination of different directions where each $\Delta\mathbf{h}_t$ corresponds to a distinct attribute, results in a generated image where all attributes have been changed.

2.2. Semantic image editing in generative models

Semantic editing has been widely explored in GANs [6, 7, 21, 25, 34, 36, 39, 40, 45]. Shen *et al*. [34] used a binary classifier to annotate generated samples and trained a SVM to separate classes like pose, age, and gender. Linear editing directions in latent space were then defined as the normal vectors of the separating hyper-planes. Härkönen *et al*. [7] found interpretable control directions in pretrained GANs by applying principal components of latent codes to appropriate layers of the generator. Another line of work [6, 36, 39, 47] uses various factorization techniques to define meaningful directions in the latent space of GANs.

Semantic image editing has also been shown in DDMs but many existing methods make adaptations to the architecture, employ text-based optimization or model fine-tuning. In DiffusionAE [22], a DDM was trained in conjunction with an image encoder. This enabled attribute manipulation on real images, including modifications of gender, age, and smile, but requires modifying the DDM architecture. Another line of work includes DiffusionCLIP [12], Imagic [11], and UniTune [42], combined CLIP-based text guidance with model fine-tuning. Unlike these methods, our approaches do not require CLIP-based text-guidance nor model fine-tuning and can be applied to existing DDMs without retraining or adapting the architecture.

We acknowledge as concurrent work the unsupervised method proposed by Park *et al*. [20]. They perform spectral analysis on the Jacobian of a mapping from pixel space to a reduced h-space consisting of the sum-pooled feature map of the bottleneck representation. In comparison, our proposed method is able to operate on the full bottleneck representation using power iteration to circumvent the intractable computational cost of calculating the Jacobian explicitly. We further propose to allow for additional region-specific control by calculating the Jacobian with respect to a region of interest, allowing for fine-grained and highly localized semantic editing.

3. The semantic latent space of DDMs

Diffusion models are defined in terms of a forward diffusion process that adds increasing amounts of white Gaussian noise to a clean image $\mathbf{x}_0$ in T steps, and a learned reverse process that gradually removes the noise. During the forward process each noisy image $\mathbf{x}_t$ is generated as

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\mathbf{n}, \tag{1}$$

where $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the noise schedule is defined by $\{\alpha_t\}$. In [38], generating an image from the model is done by first sampling Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, which is then denoised following the approximate reverse diffusion process

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\mathbf{P}_t(\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t)) + \mathbf{D}_t(\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t)) + \sigma_t \mathbf{z}_t, \tag{2}$$

where $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Here $\boldsymbol{\epsilon}_t^\theta$ is a neural network (usually a U-Net [29]), which is trained to predict $\mathbf{n}$ from $\mathbf{x}_t$, and the terms

$$\mathbf{P}_t(\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t)) = \frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t)}{\sqrt{\alpha_t}} \tag{3}$$

$$\mathbf{D}_t(\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t)) = \sqrt{1-\alpha_{t-1}-\sigma_t^2}\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t) \tag{4}$$

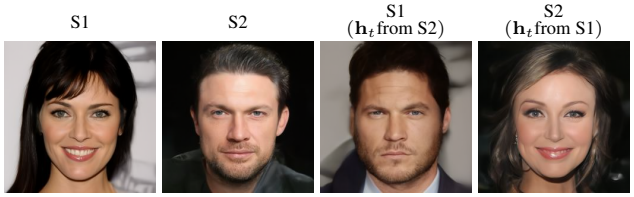
are the predicted $\mathbf{x}_0$ and the direction pointing to $\mathbf{x}_t$ at timestep t, respectively. The variance σ_t is taken to be

$$\sigma_t = \eta_t\sqrt{(1-\alpha_{t-1})/(1-\alpha_t)}\sqrt{1-\alpha_t/\alpha_{t-1}}. \tag{5}$$

The special case where $\eta_t = 0$ for all t is called DDIM [38]. In this setting the noise variance is $\sigma_t = 0$, so that the sampling process is deterministic and fully reversible [3, 9] (*i.e.,* $\mathbf{x}_T$ can be uniquely obtained from $\mathbf{x}_0$). The case where $\eta_t = 1$ corresponds to the stochastic DDPM scheme [9].

Following Kwon *et al.* [14], we study the semantic latent space of DDMs corresponding to the activation of the bottleneck feature maps of the U-Net. We denote the concatenation of the bottleneck activation across all timesteps as $\mathbf{h}_{T:1}$ see supplementary material (SM) Sec. A for illustration and additional details. In [14] image editing was performed via an asymetric reverse process (Asyrp), where $\Delta\mathbf{h}_t$ is only injected into $\mathbf{P}_t$ of (2) and not to $\mathbf{D}_t$. Empirically, we find that Asyrp amplifies the effect of the edits but semantic editing is also possible without using Asyrp. In this paper, we inject $\Delta\mathbf{h}_t$ into both terms of (2). This has the benefit of only requiring a single forward pass of the U-Net at each step of the sampling process, as opposed to the two forward passes needed in Asyrp (one for $\mathbf{P}_t$ with injection and one for $\mathbf{D}_t$ without the injection). In SM Sec. B we provide a comparison of the effect of editing with and without using Asyrp.

The bottleneck activation $\mathbf{h}_t$ is determined directly from $\mathbf{x}_t$ in each step of the generative process. It is worth noting that although most of the high-level semantic content of the generated image is determined by $\mathbf{h}_{T:1}$, it is not a complete latent representation in the sense that it does not completely specify the generated image. We illustrate this point in Fig. 2a where we swap $\mathbf{h}_{T:1}$ between two samples while keeping $\{\mathbf{x}_T, \mathbf{z}_{T:1}\}$ fixed. We observe that swapping $\mathbf{h}_{T:1}$ results in a swap of the high-level semantics, like the gender, but not the background.



(a) Effect of swapping the bottleneck activation.

(b) Vector arithmetic in the semantic latent space.

Figure 2. **Illustration of properties of the h-space.** (a) Swapping $\mathbf{h}_{T:1}$ between two samples, S1 and S2, swaps the semantic content without affecting background. (b) Adding the difference in bottleneck activation $\mathbf{h}_{T:1}$ between a smiling and non-smiling person results in a smile in a new sample. The result are shown with strength parameter $\gamma = 1/5$.

A key property of h-space is that it obeys vector arithmetic properties which have previously been demonstrated for GANs by Radford *et al.* [24]. Specifically, image editing can be done in h-space as follows. Suppose we have found a direction $\mathbf{v}_{T:1}$ associated with some semantic content that we wish to apply to a sample with latent code $\mathbf{h}_{T:1}$. Then $\mathbf{h}_{T:1}^{(\text{edit})} = \mathbf{h}_{T:1} + \gamma\mathbf{v}_{T:1}$ is the latent code of the edited image, where γ controls the strength of the edit. In Fig. 2b we illustrate the vector arithmetic property of h-space by adding a difference vector which has the semantic effect of adding a smile.

4. Unsupervised semantic directions

4.1. Principal component analysis

Our first goal is to uncover interesting semantic directions in an unsupervised fashion. To this end, we first explore the use of principal component analysis (PCA) in h-space. In the context of GANs [7], it was shown that the principal components of a collection of randomly sampled latent codes result in semantically interpretable editing directions. Here we demonstrate that the same is true for DDMs if the PCA is performed in the semantic h-space. Specifically, we consider PCA where we generate n ran-

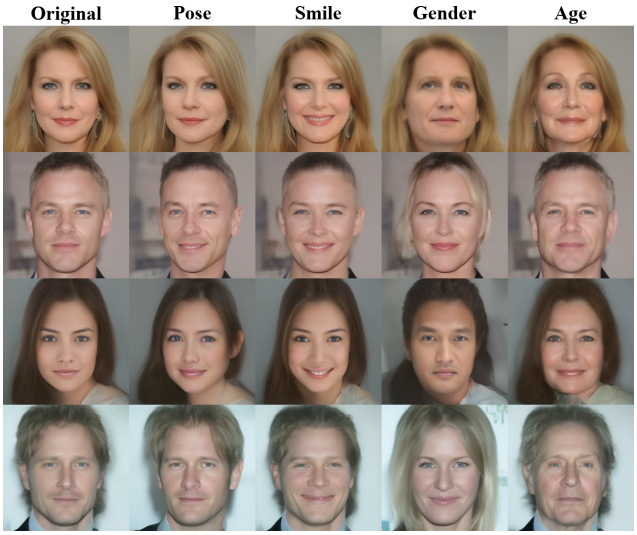
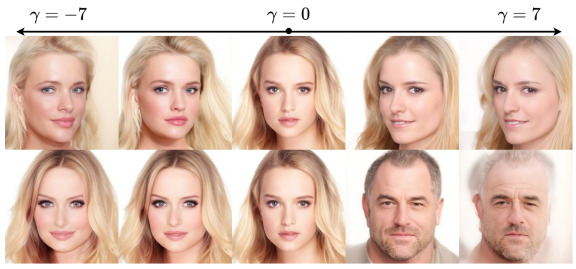


Figure 3. **PCA in the semantic latent space.** PCA in h-space provides a way for discovering disentangled and semantically meaningful directions. Here we show a selection of semantic edits corresponding to pose, smile, gender and age.



(a) Two dominant PCA directions

(b) Random directions

Figure 4. **PCA v. random directions** While directions found with PCA have a clear semantic meaning, like pose and gender, interpolating along random directions results in only minor changes to the image when using the same scale. Increasing the scale results in a degradation of the image.

dom samples and save the bottleneck activation $\mathbf{h}_t^{(i)}$ for each sample i at all timesteps. Then, for each timestep t we vectorize $\{\mathbf{h}_t^{(i)}\}_{i=1}^n$ and calculate the principal components. We define the editing direction $\mathbf{v}_j$ as a concatenation of the j'th principal component from all timesteps. To demonstrate our method, we use Diffusers [43] and a DDPM[1] trained on the CelebA [16] data set. Unless stated otherwise, all results use $\eta_t = 1$ during the synthesis process.

It can be seen that many principal directions have clear semantic interpretations, Fig. 3 demonstrates the effect of several of these directions, including directions corresponding to gender $(\mathbf{v}_1)$, pose $(\mathbf{v}_2)$, age $(\mathbf{v}_4)$, and smile $(\mathbf{v}_{10})$. Fig. 4a and 4b compares the effect of applying the two dominant principal components to applying random directions. For a fair comparison, we set the norm of $\Delta\mathbf{h}_t$ for the random directions to match that of the principal components. While interpolating along principal directions leads to semantically interpretable edits, shifting along random directions only induces minor changes to the image at small scales and rapid degradation of the image at larger scales.

4.2. Discovering image-specific semantic edits

The directions found with PCA are computed based on many samples and tend to find global changes such as pose and gender, while more local changes like the closing of the eyes are absent. The smile direction is the only direction we observed where the semantic changes are localized to a specific region like the mouth. In the following, we present a method to find directions that are specific to a single image and region of interest.

To find directions specific to a single image we wish to find a set of orthogonal directions in h-space that induce the largest change in the prediction of the clean image $\mathbf{P}_t(\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t))$ at every timestep. This is equivalent to finding the directions that change $\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t)$ the most (see SM Sec. C). For small perturbations, these directions are the top right-hand singular vectors of the Jacobian of $\boldsymbol{\epsilon}_t^\theta$ with respect to $\mathbf{h}_t$. Due to the skip-connections in the U-Net, the output of the network depends on both $\mathbf{x}_t$ and $\mathbf{h}_t$. Yet, here we only consider the dependency on the latent variable $\mathbf{h}_t$. In the following, we denote the Jacobian of $\boldsymbol{\epsilon}_t^\theta$ by $\mathbf{J}_t$ and its singular value decomposition (SVD) as

$$\mathbf{J}_t \triangleq \frac{\partial \boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t, \mathbf{h}_t)}{\partial \mathbf{h}_t} = \mathbf{U}_t \boldsymbol{\Sigma}_t \mathbf{V}_t^{\mathrm{T}}. \tag{6}$$

The right singular vectors corresponding to the largest singular values, (the columns of $\mathbf{V}_t$) are the set of orthogonal vectors in h-space which perturb the predicted image the most. Note that for each timestep t, we have a different set of directions. In practice, we find that semantically interesting effects are obtained by applying directions found at timestep t across all timesteps. Thus, computing k directions per timestep provide us kT potential edits in each of the T timesteps. In SM Sec. D, we illustrate the qualitative difference between directions computed at different timesteps.

[1] https://huggingface.co/google/ddpm-ema-celebahq-256

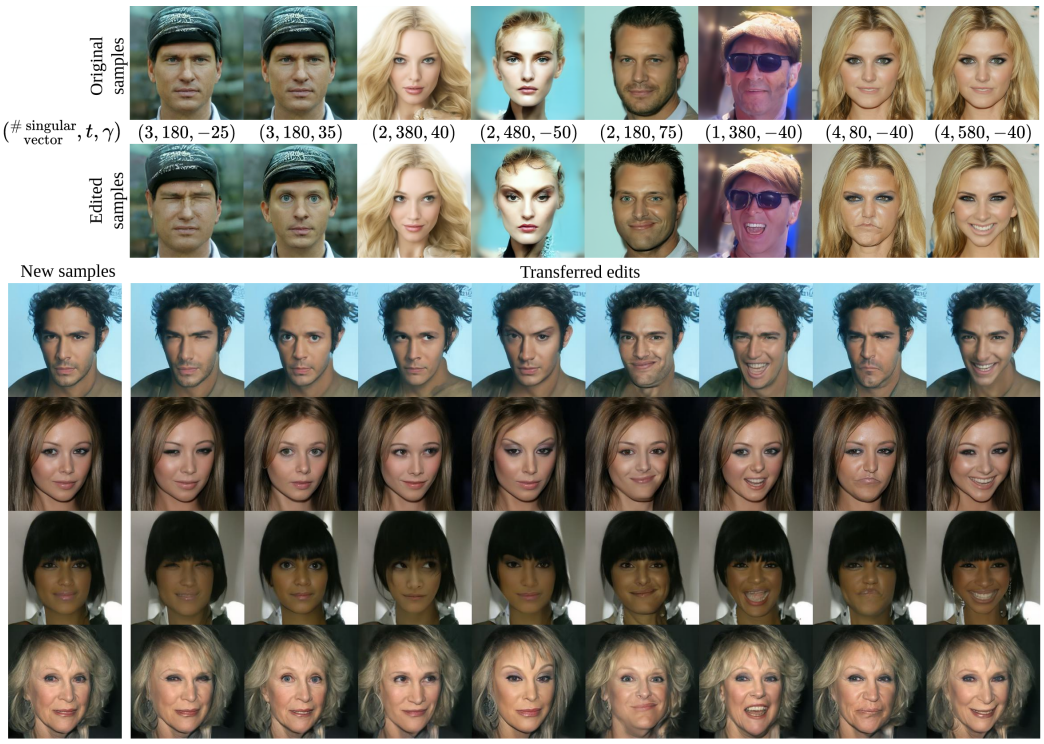


Figure 5. **Unsupervised image-specific edits.** Spectral analysis of the Jacobian of $\boldsymbol{\epsilon}_t^\theta$ yields directions corresponding to localized changes in the generated image, *e.g.* eyes opening/closing and raising of the eyebrows. Although this method is image-specific, directions found for one sample can be transferred to others, where they result in semantically similar edits.

In practice, calculating $\mathbf{J}_t$ directly is computationally expensive. Instead, we find the dominant singular vectors by power-iteration over the matrix $\mathbf{J}_t^{\mathrm{T}}\mathbf{J}_t$, whose eigenvectors are precisely the right singular vectors of $\mathbf{J}_t$. Each iteration requires multiplication by $\mathbf{J}_t^{\mathrm{T}}\mathbf{J}_t$, which can be computed without ever storing the Jacobian matrix in memory. Specifically, for any vector $\mathbf{v}$, the product $\mathbf{J}_t^{\mathrm{T}}\mathbf{J}_t\mathbf{v}$ can be computed as

$$\mathbf{J}_t^{\mathrm{T}}\mathbf{J}_t\mathbf{v} = \frac{\partial}{\partial \mathbf{h}_t}\left\langle \boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t, \mathbf{h}_t), \mathbf{J}_t\mathbf{v}\right\rangle \tag{7}$$

with

$$\mathbf{J}_t\mathbf{v} = \frac{\partial}{\partial a}\boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t, \mathbf{h}_t + a\mathbf{v})\Big|_{a=0}. \tag{8}$$

Our algorithm is summarized in Alg. 1 and uses (7) to calculate the singular vectors of the Jacobian of an arbitrary vector-valued function $\mathbf{f}$. The algorithm starts by randomly initializing a set of vectors $\{\mathbf{v}_i\}_{i=1}^k$ and iterative computes (7) using automatic differentiation while enforcing orthogonality among the singular vectors. Importantly, it was shown that batched power iteration with an orthogonalization step, such as presented here, is guaranteed to converge to the SVD of positive semi-definite matrices [32, Ch. 5].

Algorithm 1 Jacobian subspace iteration

Input: $\mathbf{f} : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{d_{\text{out}}}$, $\mathbf{h} \in \mathbb{R}^{d_{\text{in}}}$ and $\mathbf{V} \in \mathbb{R}^{d_{\text{in}} \times k}$
Output: $(\mathbf{U}, \boldsymbol{\Sigma}, \mathbf{V}^{\mathrm{T}})$ – k largest singular values and singular vectors of the Jacobian $\partial\mathbf{f}/\partial\mathbf{h}$

```
y ← f(h)
if V is empty then
    V ← i.i.d. standard Gaussian samples
end if
Q, R ← QR(V)                          ▷ Reduced QR decomposition
V ← Q                                 ▷ Ensures V^T V = I
while stopping criteria do
    U ← ∂f(h + aV)/∂a at a = 0        ▷ Batch forward
    V̂ ← ∂(U^T y)/∂h
    V, Σ², R ← SVD(V̂)                 ▷ Reduced SVD
end while
Orthonormalize U
```

Regarding implementation, in (7) we compute a derivative of high dimensional output w.r.t. a scalar. This is efficiently done by utilizing forward mode automatic differentiation. Further, (7) can be calculated in parallel for multiple vectors using the batched Jacobian-vector product, *e.g.* in PyTorch. However, parallel calculation of a large num-

Figure 6. **Region-specific edits.** Given a mask specifying a region of interest, our method can be guided to focus on finding directions which change only the target area. The first column shows the input with the mask shown in green.

ber of vectors can be memory intensive. For such cases, we give a sequential variant of Alg.1 in SM, Sec. E.

Our proposed method successfully identifies semantically meaningful directions that correspond to highly localized semantic changes in the image, such as closing or opening of the eyes and mouth, or raising of the eyebrows. We show a selection of such localized edits at the top of Fig. 5. While the semantic directions found by this method are image-specific and may vary depending on the sample analyzed, we find that they result in the same localized changes when applied across different images. This is illustrated in the lower part of Fig. 5 where each of the found editing directions is applied with the same magnitude γ across a selection of samples. These results suggest that our approach is effective in identifying meaningful semantic directions that generalize across different images.

If additional information is available in the form of a mask specifying a region of interest, our method can be naturally extended by applying the mask to the noise prediction $\widetilde{\boldsymbol{\epsilon}}_t^\theta$ in order to find directions in h-space that change a specific region the most rather than the whole image. We seek the singular vectors of the Jacobian of the masked output of the U-net. We define the a masked Jacobian $\mathbf{J}_t^{\text{masked}}$ as

$$\mathbf{J}_t^{\text{masked}} = \partial \widetilde{\boldsymbol{\epsilon}}_t^\theta(\mathbf{x}_t, \mathbf{h}_t)/\partial \mathbf{h}_t, \tag{9}$$

with $\widetilde{\boldsymbol{\epsilon}}_t^\theta(\mathbf{x}_t, \mathbf{h}_t) = \boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t, \mathbf{h}_t) \odot \mathbf{M}$, where $\odot$ denotes the Hadamard product and $\mathbf{M}$ is a binary mask corresponding to a region of interest. We show examples of such region-specific edits in Fig. 6 where we apply masking to discover editing directions corresponding to changes to the eyes, hair and eyebrows.

5. Supervised discovery of semantic directions

While the methods we presented in Sec. 4 discover interpretable semantic directions in a fully unsupervised fashion, their effects must be interpreted manually. In this section, we demonstrate a simple supervised approach to obtain latent directions corresponding to well-defined labels.

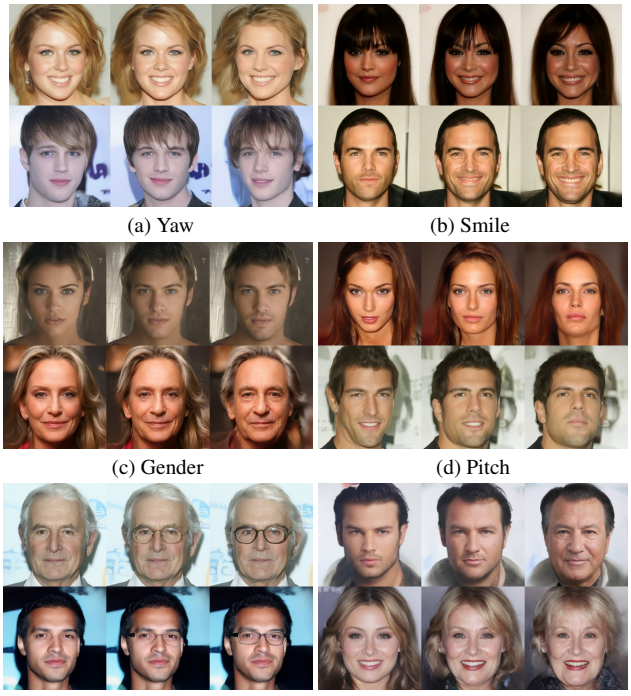
Linear semantic directions from examples. The vector arithmetic property of h-space suggests an intuitive method for discovering semantically meaningful directions, by providing positive and negative examples of a desired attribute. Let $\{(\mathbf{x}_i^-, \mathbf{x}_i^+)\}_{i=1}^n$ be a collection of generated images, such that all $\mathbf{x}_i^+$ have a desired attribute that is absent in $\mathbf{x}_i^-$, *e.g.* a smile, old age, glasses, *etc*. Let $\mathbf{q}_i^-$ and $\mathbf{q}_i^+$ denote the latent representation corresponding to the images $\mathbf{x}_i^-$ and $\mathbf{x}_i^+$. Then, we can find a semantic direction $\mathbf{v}$ as

$$\mathbf{v} = \frac{1}{n} \sum_{i=1}^{n} \left(\mathbf{q}_i^+ - \mathbf{q}_i^- \right). \tag{10}$$

Note that this method can be applied using either $\mathbf{h}_{T:1}$ or $\mathbf{x}_T$ as the latent variable. However, defining semantic directions using $\mathbf{h}_{T:1}$ as the latent variable requires far fewer samples than using $\mathbf{x}_T$. Figure 8a illustrates this for DDIM ($\eta_t = 0$) for a direction corresponding to smile where (10) is calculated using a varying number of samples.

Classifier annotation. We now propose to find linear semantic directions by using pretrained attribute classifiers to annotate samples generated by the model. Using the attribute classifier from [15], we annotate samples with probabilities corresponding to the 40 classes from CelebA [16], and use Hopenet [30] to predict pose (yaw, pitch, and roll). We sort the annotated samples according to the attribute scores and select the samples with the highest and lowest scores from each class as the positive and negative examples respectively. We then calculate semantic directions corresponding to the different attributes using the method given in (10).

As shown in Fig. 7, we can successfully find semantic directions controlling a wide selection of meaningful attributes like yaw, smile, gender, glasses, and age. Furthermore, directions calculated by (10) can be applied in combination with one another. For example, adding $\Delta\mathbf{h}_{T:1}$ for two attributes, like pose and smile, results in an image where both attributes are changed. Figure 8b illustrates sequential editing, showcasing changes in expression followed by pose, age, and eyeglasses for two samples. In SM Sec. F we show that this method can be applied to find directions corresponding to facial expressions using DDIM



(a) Yaw (b) Smile

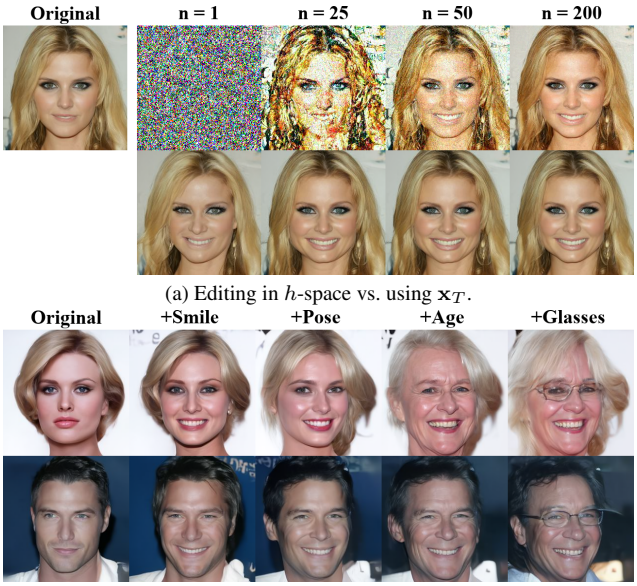
(c) Gender (d) Pitch

(e) Glasses (f) Age

Figure 7. **Single attribute manipulation.** Using a domain-specific binary attribute classifier, we find linear directions in h-space corresponding to a variety of semantic edits.

inversion and a real facial expression data set [46] as supervision.

Disentanglement of semantic directions. Latent directions found by (10) might be semantically entangled, in the sense that editing in the direction corresponding to some desired attribute might also induce a change in some other undesired attributes. For example, a direction for eyeglasses may also affect the age if it correlates with eyeglasses in the training data. To remedy this, we propose conditional manipulation in h-space in a way similar to what was suggested in the context of GANs by Shen *et al*. [34, 35]. Let $\mathbf{v}_1$ and $\mathbf{v}_2$ be two linear semantic directions, where the two corresponding semantic attributes are entangled. We can define a new direction $\mathbf{v}_{1\perp 2}$ which only affects the semantics associated with $\mathbf{v}_1$, without changing the semantics associated with $\mathbf{v}_2$. This is done simply by removing from $\mathbf{v}_1$ the projection of $\mathbf{v}_1$ onto $\mathbf{v}_2$, namely $\mathbf{v}_{1\perp 2} = \mathbf{v}_1 - \langle \mathbf{v}_1, \mathbf{v}_2 \rangle / \|\mathbf{v}_2\|^2 \mathbf{v}_2$. In case of conditioning on multiple semantics simultaneously, our aim is to remove the effects of a collection of k directions $\{\mathbf{v}_i\}_{i=1}^k$ from a primal direction $\mathbf{v}_0$ in order to define a new direction $\mathbf{v}$ which only affects the target attribute. This can be done by concatenating the k directions into a matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_k]$ and projecting $\mathbf{v}_0$ onto the orthogonal complement of the



(a) Editing in h-space vs. using $\mathbf{x}_T$.

(b) Sequential manipulation.

Figure 8. **Editing properties of h-space.** (a) A qualitative comparison of the editing effect using $\mathbf{x}_T$ (top) and $\mathbf{h}_{T:1}$ (bottom) as the latent variables using a smiling direction found by (10). While the direction in h-space converges with a few labeled examples, more than 200 are required to achieve a similar result using $\mathbf{x}_T$ as the latent variable. (b) Directions found with our method can be applied in combination with one another. Here, we sequentially accumulate four effects, starting from a single effect in the second column up to four effects in the fifth column.

column space of the matrix $\mathbf{V}$ using

$$\mathbf{v} = \left[\mathbf{I} - \mathbf{V}\left(\mathbf{V}^{\mathrm{T}}\mathbf{V}\right)^{-1}\mathbf{V}^{\mathrm{T}}\right]\mathbf{v}_0. \tag{11}$$

The resulting direction will be disentangled from each of the directions $\{\mathbf{v}_i\}$, meaning that moving a sample along this new direction will result in a large change in the attribute associated with $\mathbf{v}_0$ while minimally affecting the attributes associated with the other directions. Figure. 9 visualizes the effect of interpolating in the directions of age and eyeglasses for two samples. As can be seen, these directions are entangled with gender and age, respectively. By using our method we can successfully remove the entanglement and define a direction which only affects age or the presence of glasses.

To validate the effectiveness of our disentanglement strategy, we performed an experiment where we edited attributes corresponding to smile, glasses, age, gender, and wearing a hat. We edited samples using both the original and the disentangled directions while measuring the effect of each edit using CLIP [23] as a zero-shot classifier. We selected appropriate positive and negative prompts for each attribute. For smiling, glasses, and hat we used `"A smiling person"`, `"A person wearing`

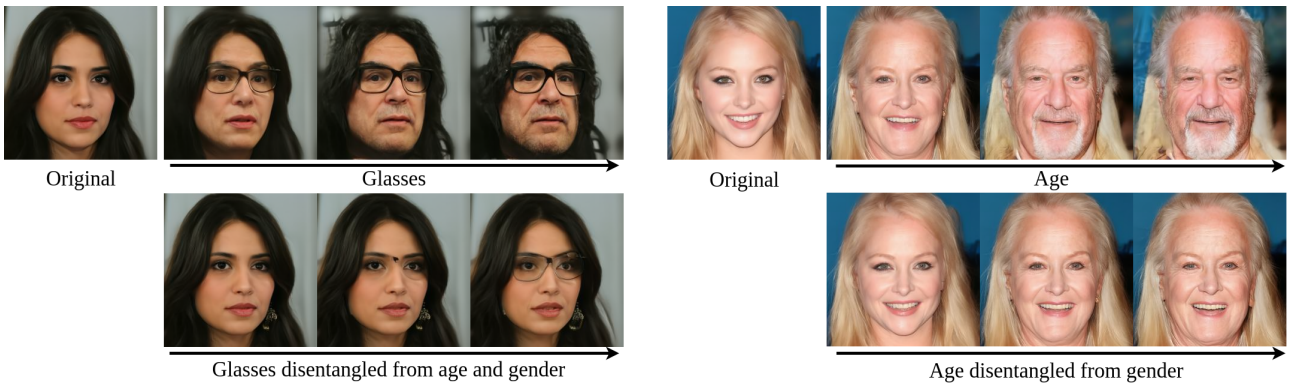


Figure 9. **Disentanglement of semantic directions.** Given a direction that is entangled with other attributes, we can create a disentangled direction by removing the projection onto undesired semantics. The top row shows the original direction, whereas the bottom row shows the disentangled direction.

Table 1. **Evaluation of disentanglement strategy.** We quantitatively evaluate the effect of disentangling semantic directions using linear projection. The rows correspond to the applied directions, while the columns correspond to the effect of the edits according to CLIP. We draw and edit 100 random samples and repeat the experiment 10 times with different seeds and report the mean and standard deviations. The strongest effect in each row is highlighted.

Edit \ Effect	Smile	Glasses	Age	Gender	Hat	Smile	Glasses	Age	Gender	Hat
	Original directions					Disentangled directions				
Smile	0.26±0.02	0.29±0.02	0.08±0.02	**0.31±0.04**	0.07±0.01	**0.24±0.02**	0.20±0.02	0.04±0.02	0.09±0.03	0.03±0.01
Glasses	0.48±0.02	0.32±0.02	**0.68±0.03**	0.66±0.04	0.14±0.02	0.22±0.01	**0.38±0.02**	0.13±0.02	0.07±0.03	0.36±0.02
Age	0.07±0.01	0.40±0.03	**0.74±0.03**	0.66±0.04	0.18±0.01	0.02±0.02	0.38±0.03	**0.59±0.04**	0.16±0.03	0.04±0.02
Gender	0.40±0.02	0.28±0.03	0.58±0.03	**0.66±0.04**	0.09±0.02	0.20±0.02	0.01±0.01	0.08±0.02	**0.39±0.03**	0.07±0.02
Hat	0.42±0.02	0.39±0.02	0.37±0.03	**0.66±0.04**	0.41±0.02	0.13±0.01	0.03±0.03	0.02±0.03	0.02±0.09	**0.44±0.02**

`glasses"` and `"A person wearing a hat"` for the positive prompts respectively, and `"A person"` as the negative prompt. For age and gender, we used `"A man"` / `"A woman"` and `"An old person"` / `"A young person"` respectively. For each sample, we edited each of the five attributes and measured the change in attribute score according to CLIP. Table 1 shows the results. We can see that the original directions are highly entangled with other attributes while the disentangled directions induce the largest changes in the intended attributes. This demonstrates that semantic directions can be disentangled by a simple linear projection.

6. Discussion and conclusion

We presented several supervised and unsupervised methods for finding interpretable directions in the recently proposed semantic latent space of Denoising Diffusion Models. We showed that the principal components in latent space correspond to global and semantically meaningful editing directions like pose, gender, and age. Additionally, we proposed a novel method for discovering directions based on a single input image. These directions correspond to highly localized changes in generated images, such as raising the eyebrows or opening/closing the mouth and eyes. Although these directions were found with respect to a specific image they can be transferred to different samples.

As our proposed methods enable high-quality editing of face images, we provide a broader impact statement in SM Sec G. Although our unsupervised approaches are effective in discovering meaningful semantics when the DDM was trained on aligned data like human faces, we found that models trained on less structured data have less interpretable principal directions. We refer the reader to SM Sec. H for experiments on models trained on churches and bedrooms.

Further, we proposed a conceptually simple supervised method utilizing the linear properties of the semantic latent space. We showed that a diverse set of face semantics can be revealed using an attribute classifier to annotate samples. Finally, we demonstrated that simple linear projection is an effective strategy for disentangling otherwise correlated semantic directions. All of our proposed methods apply to pretrained DDMs without requiring any adaptation to the model architecture, fine-tuning, optimization, or text-based guidance.

References

[1] Yuval Alaluf, Or Patashnik, Zongze Wu, Asif Zamir, Eli Shechtman, Dani Lischinski, and Daniel Cohen-Or. Third time's the charm? image and video editing with stylegan3. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 204–220. Springer, 2023. 1

[2] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022. 1

[3] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021. 1, 3

[4] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 1

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, page 2672–2680. Curran Associates, Inc., 2014. 1

[6] René Haas, Stella Graßhof, and Sami Sebastian Brandt. Tensor-based emotion editing in the stylegan latent space. *arXiv:2205.06102 [cs]*, May 2022. Accepted for poster presentation at AI4CC @ CVPRW. 1, 2

[7] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *Advances in Neural Information Processing Systems*, 33:9841–9850, 2020. 1, 2, 3

[8] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 1

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. 3

[10] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 1

[11] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Conference on Computer Vision and Pattern Recognition 2023*, 2023. 1, 2

[12] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2426–2435, June 2022. 1, 2

[13] Gihyun Kwon and Jong Chul Ye. Diffusion-based image translation using disentangled style and content representation. *arXiv preprint arXiv:2209.15264*, 2022. 1

[14] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. In *International Conference on Learning Representations*, 2023. 2, 3, 12

[15] Ji Lin, Richard Zhang, Frieder Ganz, Song Han, and Jun-Yan Zhu. Anycost gans for interactive image synthesis and editing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 6

[16] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 4, 6

[17] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. 1, 19

[18] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *arXiv preprint arXiv:2211.09794*, 2022. 1

[19] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pages 16784–16804. PMLR, 2022. 1

[20] Yong-Hyun Park, Mingi Kwon, Junghyo Jo, and Youngjung Uh. Unsupervised discovery of semantic latent directions in diffusion models. *arXiv preprint arXiv:2302.12469*, 2023. 2

[21] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2085–2094, October 2021. 2

[22] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 10609–10619, New Orleans, LA, USA, Jun 2022. IEEE. 2

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and et al. Learning transferable visual models from natural language supervision. In *Proc. ICML*, Feb 2021. arXiv: 2103.00020. 1, 7

[24] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. 3

[25] Yipeng Qin Rameen Abdal and Peter Wonka. Image2StyleGAN++: How to edit the embedded images? In *Proc. CVPR*, pages 8293–8302, Aug 2020. 1, 2

[26] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022. 1

[27] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021. 1

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 1

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. 2, 3

[30] Nataniel Ruiz, Eunji Chong, and James M. Rehg. Fine-grained head pose estimation without keypoints. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 6

[31] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2208.12242*, 2022. 1

[32] Yousef Saad. *Numerical methods for large eigenvalue problems: revised edition*. SIAM, 2011. 5

[33] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 1

[34] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020. 1, 2, 7

[35] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *TPAMI*, 2020. 7

[36] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *CVPR*, 2021. 1, 2

[37] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. 1

[38] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, October 2020. 2, 3

[39] Nurit Spingarn, Ron Banner, and Tomer Michaeli. GAN Steerability without optimization. In *International Conference on Learning Representations*, 2021. 1, 2

[40] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zöllhofer, and Christian Theobalt. StyleRig: Rigging StyleGAN for 3d control over portrait images. In *Proc. CVPR)*. IEEE, June 2020. 2

[41] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. *arXiv preprint arXiv:2211.12572*, 2022. 1

[42] Dani Valevski, Matan Kalman, Yossi Matias, and Yaniv Leviathan. Unitune: Text-driven image editing by fine tuning an image generation model on a single image, 2022. 2

[43] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 4

[44] Xin Wang, Hui Guo, Shu Hu, Ming-Ching Chang, and Siwei Lyu. Gan-generated faces detection: A survey and new perspectives. *ArXiv*, abs/2202.07145, 2022. 19

[45] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for StyleGAN image generation. In *Proc. CVPR*, Dec 2020. 2

[46] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and M.J. Rosato. A 3d facial expression database for facial behavior research. In *7th Intern. Conf. on Automatic Face and Gesture Recognition (FGR06)*, pages 211–216, 2006. 7, 19

[47] Jiapeng Zhu, Ruili Feng, Yujun Shen, Deli Zhao, Zhengjun Zha, Jingren Zhou, and Qifeng Chen. Low-rank subspaces in GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

Supplemental Materials

A. Illustration of h-space.

In this paper, we define h-space as the space of bottleneck activations $\mathbf{h}_t$ across each of the T timesteps in the synthesis process. See illustration in Fig.10. Each downsampling block increases the number of channels while decreasing the spacial dimension of the feature maps. In our case, using the pretrained DDPM model trained on CelebA released by Google[2]. The input pixel space has dimensions $(3, 256, 256)$ and the deepest feature map has dimensions $(512, 8, 8)$. Thus an element of h-space, $\mathbf{h}_{T:1}$, has dimensions $(T, 512, 8, 8)$ and is defined as

$$\mathbf{h}_{T:1} = \mathbf{h}_T \otimes \mathbf{h}_{T-1} \otimes \cdots \otimes \mathbf{h}_2 \otimes \mathbf{h}_1. \tag{12}$$

We apply directions in h space by perturbing $\mathbf{h}_{T:1}$ with some offset as $\mathbf{h}_{T:1} + \Delta\mathbf{h}_{T:1}$ during the generative process in (2). When $\eta_t \neq 0$ the clean image is completely specified by the triple $(\mathbf{x}_T, \mathbf{z}_{T:1}, \Delta\mathbf{h}_{T:1})$ and for $\eta_t = 0$ (DDIM) it is determined by the tuple $(\mathbf{x}_T, \Delta\mathbf{h}_{T:1})$.

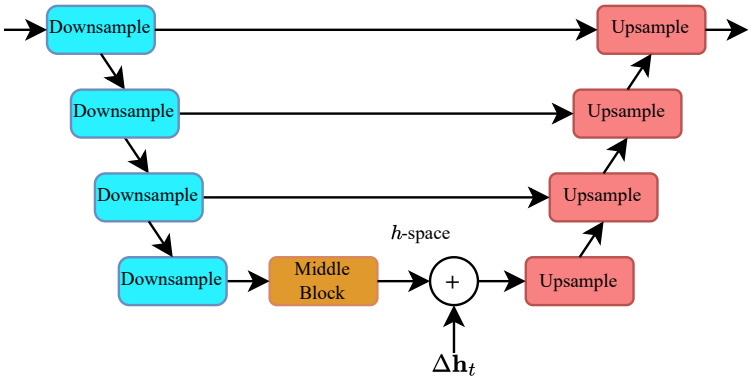
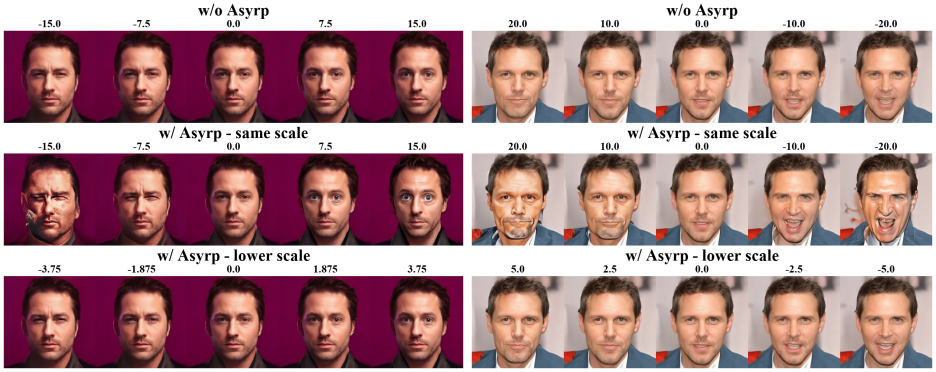


Figure 10. **Illustration of h-space.** In this paper, we define the semantic latent space of DDMs as the activation after the deepest bottleneck layer of the U-Net.

[2] https://huggingface.co/google/ddpm-ema-celebahq-256

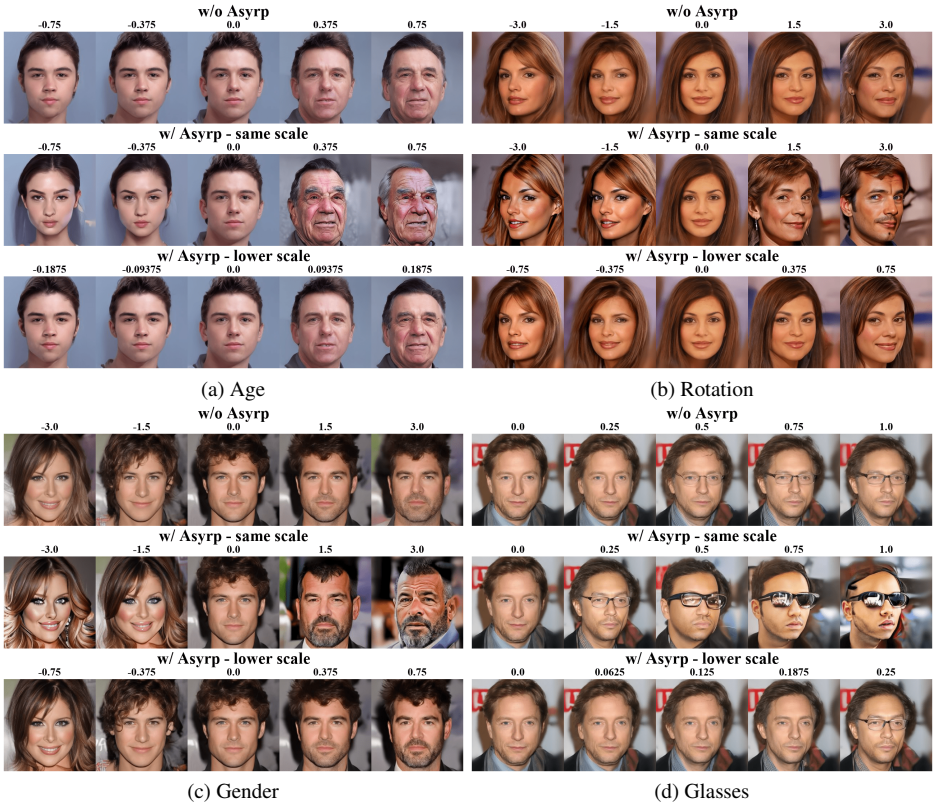
B. The effect of Asyrp

In the main text, we stated that using Asyrp [14] acts to amplify the effect edits in h-space. However, Asyrp is computationally costly since it requires two forward passes of the U-Net at each denoising step. Hence, Asyrp is not used for any of the results shown in the main paper. In Figs. 11 and 12 we qualitatively compare edits with and without using Asyrp. We observe that simply adjusting the scale of the applied direction results in very similar edits.



(a) Eyes (b) Mouth

Figure 11. **The Effect of Asyrp.** Results are shown for directions found with Alg. 1.



(a) Age (b) Rotation

(c) Gender (d) Glasses

Figure 12. **The effect of Asyrp.** Results are shown for directions found using the supervised method presented in Sec. 5.

C. A Note on image-specific directions

In the main paper, we state that the right singular vectors of the Jacobian of $\boldsymbol{\epsilon}_t^\theta$ with respect to h-space, denoted as $\mathbf{J}_t$, are the set of orthogonal vectors in h-space which perturb the noise prediction $\boldsymbol{\epsilon}_t^\theta$ the most. An equivalent statement is that those right singular vectors perturb the predicted image $\mathbf{P}_t(\mathbf{x}_t, \mathbf{h}_t)$ at timestep t the most. Specifically, since

$$\mathbf{P}_t(\mathbf{x}_t, \mathbf{h}_t) = \frac{\mathbf{x}_t - \sqrt{1-\alpha_t}}{\sqrt{\alpha_t}} \boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t, \mathbf{h}_t) \tag{13}$$

we have that

$$\frac{\partial}{\partial \mathbf{h}_t} \mathbf{P}_t(\mathbf{x}_t, \mathbf{h}_t) = -\frac{\sqrt{1-\alpha_t}}{\sqrt{\alpha_t}} \frac{\partial}{\partial \mathbf{h}_t} \boldsymbol{\epsilon}_t^\theta(\mathbf{x}_t, \mathbf{h}_t) = -\frac{\sqrt{1-\alpha_t}}{\sqrt{\alpha_t}} \mathbf{J}_t. \tag{14}$$

Thus, the eigenvectors of $(\partial \mathbf{P}_t / \partial \mathbf{h}_t)^{\mathrm{T}} (\partial \mathbf{P}_t / \partial \mathbf{h}_t)$ and $\mathbf{J}_t^{\mathrm{T}} \mathbf{J}_t$ are the same with the same ordering.

D. Image-specific directions at different timesteps

Our proposed image-specific unsupervised method in Alg. 1 finds different directions for each timestep. In Figures 13, 14, 15 and 16 we show the effect of the three dominant directions (the three top singular vectors of the Jacobian) at different timesteps along the reverse diffusion process.

Figure 13. **Directions found by Alg. 1.**

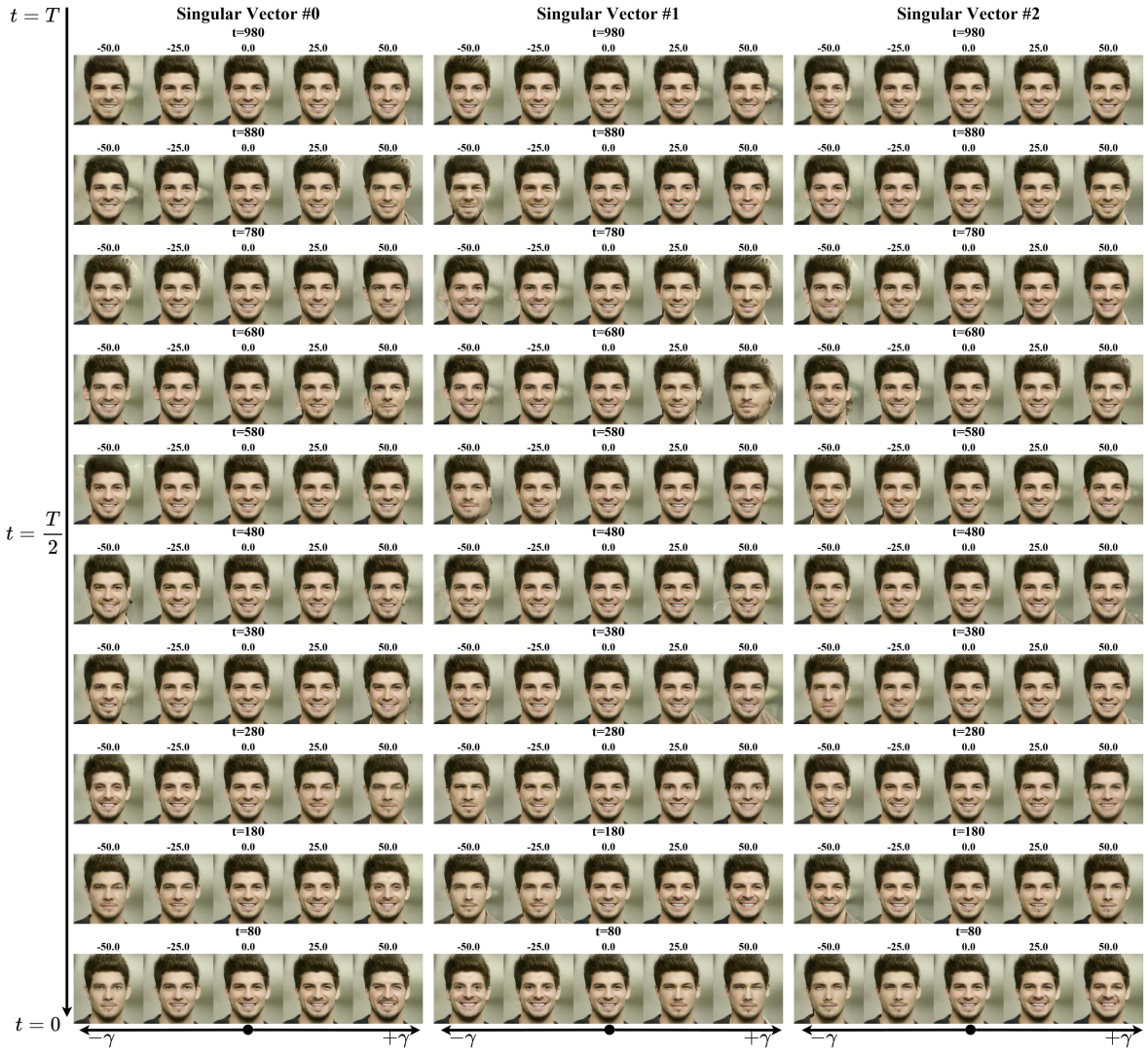


Figure 14. **Directions found by Alg. 1.**

Figure 15. **Directions found by Alg. 1.**

Figure 16. **Directions found by Alg. 1.**

E. Sequential algorithm for Jacobian subspace iteration

As mentioned in the main text, Alg. 1 can be memory intensive when calculating a large number of singular vectors in parallel. In cases where limited memory is available, we provide an alternative sequential version of our method in Alg. 2. Here we calculate the singular values and vectors in mini-batches of size b. The value of b should be set according to the parallel computation capacity. For example, in the special case of $b = 1$, the algorithm computes the vectors one by one and will use small memory. Note that lowering the mini-batch size b comes at the expense of longer running time.

Algorithm 2 Sequential Jacobian subspace iteration

Input: function to differentiate $\mathbf{f} : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{d_{\text{out}}}$, point at which to differentiate $\mathbf{h} \in \mathbb{R}^{d_{\text{in}}}$, initial guess $\mathbf{\Theta} \in \mathbb{R}^{d_{\text{in}} \times k}$ [optional], mini-batch size $b < k$

Output: $(\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}^{\mathrm{T}})$ – k top singular values and vectors of the Jacobian $\partial \mathbf{f} / \partial \mathbf{h}$

Initialization: $\mathbf{y} \leftarrow \mathbf{f}(\mathbf{h})$, $i_{\text{start}} \leftarrow 1$, $i_{\text{end}} \leftarrow b$, $\mathbf{V} \leftarrow [\,]$, $\mathbf{\Sigma} \leftarrow [\,]$, $\mathbf{U} \leftarrow [\,]$

while $i_{\text{start}} \leq k$ **do**

  if $\mathbf{\Theta}$ is empty **then**

    $\mathbf{\Phi} \leftarrow$ i.i.d. standard Gaussian samples in $\mathbb{R}^{d_{\text{in}} \times (i_{\text{end}} - i_{\text{start}} + 1)}$

  else

    $\mathbf{\Phi} \leftarrow$ columns i_{start} to i_{end} of $\mathbf{\Theta}$

  end if

  $\mathbf{Q}, \mathbf{R} \leftarrow \mathrm{QR}(\mathbf{\Phi})$ ▷ Reduced QR decomposition

  $\mathbf{\Phi} \leftarrow \mathbf{Q}$ ▷ Ensures $\mathbf{\Phi}^{\mathrm{T}} \mathbf{\Phi} = \mathbf{I}$

  while stopping criterion **do**

    if $\mathbf{V}$ is not empty **then**

      $\mathbf{\Phi} \leftarrow \left[\mathbf{I} - \mathbf{V} \left(\mathbf{V}^{\mathrm{T}} \mathbf{V}\right)^{-1} \mathbf{V}^{\mathrm{T}}\right] \mathbf{\Phi}$

      $\mathbf{\Phi}, \mathbf{R} \leftarrow \mathrm{QR}(\mathbf{\Phi})$ ▷ Reduced QR decomposition

    end if

    $\mathbf{\Psi} \leftarrow \partial \mathbf{f}(\mathbf{h} + a\mathbf{\Phi}) / \partial a$ at $a = 0$ ▷ Batch forward

    $\hat{\mathbf{\Phi}} \leftarrow \partial (\mathbf{\Psi}^{\mathrm{T}} \mathbf{y}) / \partial \mathbf{h}$

    $\mathbf{\Phi}, \mathbf{S}, \mathbf{R} \leftarrow \mathrm{SVD}(\hat{\mathbf{\Phi}})$ ▷ Reduced SVD

  end while

  $\mathbf{V} \leftarrow [\mathbf{V}; \mathbf{\Phi}]$

  $\mathbf{\Sigma} \leftarrow \begin{bmatrix} \mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}^{1/2} \end{bmatrix}$

  $\mathbf{U} \leftarrow [\mathbf{U}; \mathbf{\Psi}]$

  $i_{\text{start}} \leftarrow i_{\text{start}} + b$

  $i_{\text{end}} \leftarrow \min\{i_{\text{end}} + b, k\}$

end while

Orthonormalize $\mathbf{U}$

F. Facial expressions from real data.

We conducted an additional experiment where domain-specific semantic directions were extracted using real images as supervision. We wish to find directions corresponding to expressions like happiness, sadness, and surprise. Here we used the BU3DFE data set [46]. BU3DFE contains real images of 100 subjects, each performing a neutral expression in addition to each of the prototypical facial expressions at various intensity levels. Using DDIM inversion ($\eta_t = 0$) we recorded $\mathbf{h}_{T:1}$ during the inversion process and used (10) to calculate directions. We used the most intense expressions for the positive examples and the neutral expressions for the negative examples. The effect of the directions found using our method is shown in Fig. 17. The extracted directions are shown on generated samples. The figure shows that latent directions in h-space can successfully be found by applying our supervised method presented in Sec. 5 on a dataset of real images.

Figure 17. **Facial expressions from real data.** We extract semantic directions corresponding to different facial expressions using a data set of real images. The directions are calculated via DDIM inversion and applied in the semantic h-space to synthetic images.

G. Broader impact

In this paper, we have introduced several techniques for semantic editing of human faces using DDMs. While the creation of high-quality edited images that are difficult to distinguish from real images has significant positive applications, there is also the potential for malicious or misleading use, such as in the creation of deepfakes. Although some research has focused on detecting and mitigating the risk of AI-edited images, these have mostly focused on GANs [44] and, so far, there has been little research into detecting images that have been edited using DDMs. Given the differences in the generative process between DDMs and GANs, methods which are effective in detecting images edited by GANs might not be as effective for images edited by DDMs [17]. Further research is needed to develop effective methods for forensic analysis of edits using DDMs. Such research could help address the risk of malicious use of image-editing technologies.

H. Unsupervised methods on other domains

In addition to the model[3] trained on CelebA, which is used throughout the main paper, we also conducted experiments with models trained on churches[4] and bedrooms[5]. Although the unsupervised directions found with both PCA and Alg. 1 on these models lead to various changes to the images, these directions are less interpretable than those obtained for faces in the main paper. We showcase the first 5 PCA directions on the models trained on churches and bedrooms in Figures 18 and 19 and directions found using Alg. 1 in Figures 21 and 20.

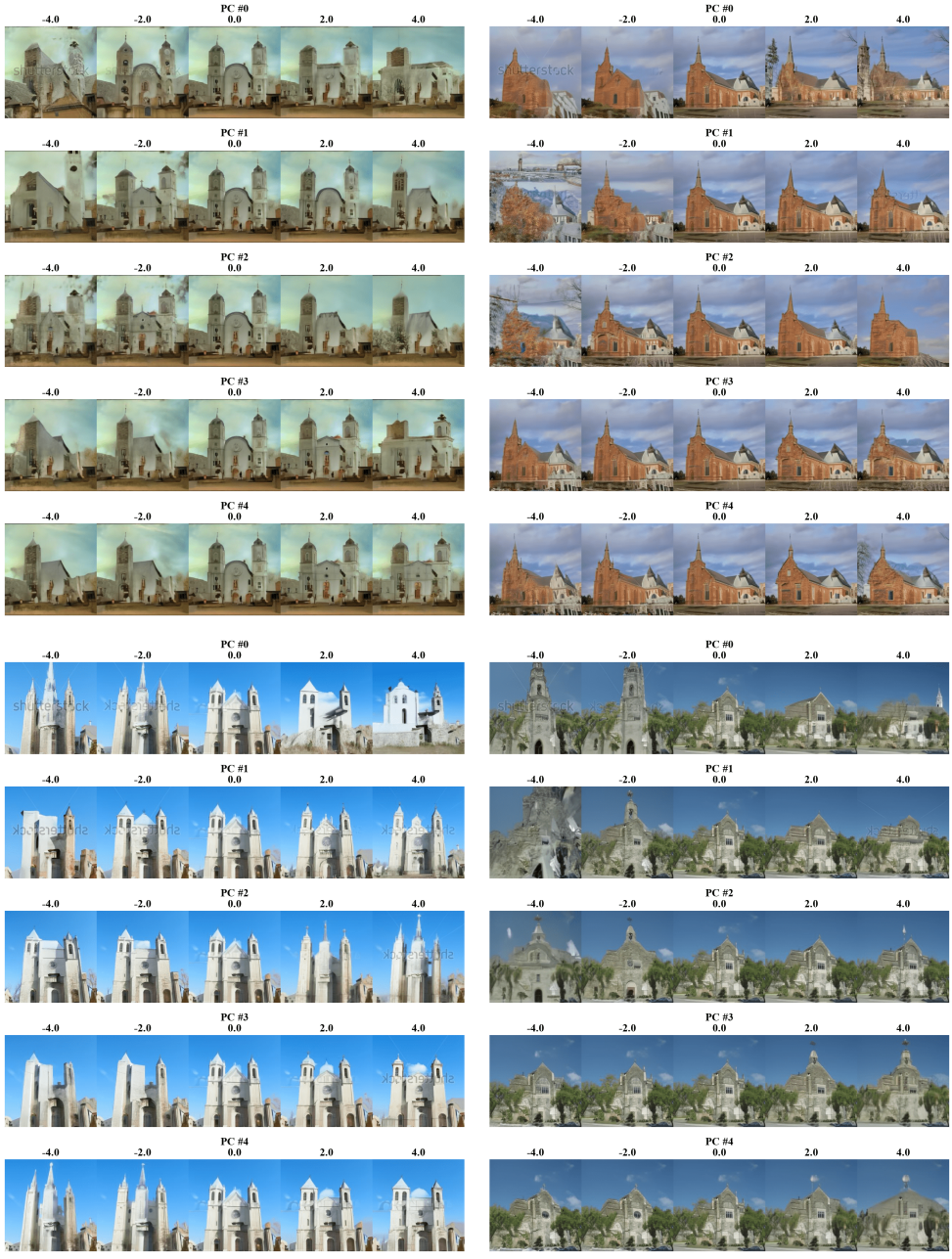


Figure 18. **PCA directions.** For a DDM trained on churches.

[3] https://huggingface.co/google/ddpm-ema-celebahq-256
[4] https://huggingface.co/google/ddpm-ema-church-256
[5] https://huggingface.co/google/ddpm-ema-bedroom-256

Figure 19. **PCA directions.** For a DDM trained on bedrooms.

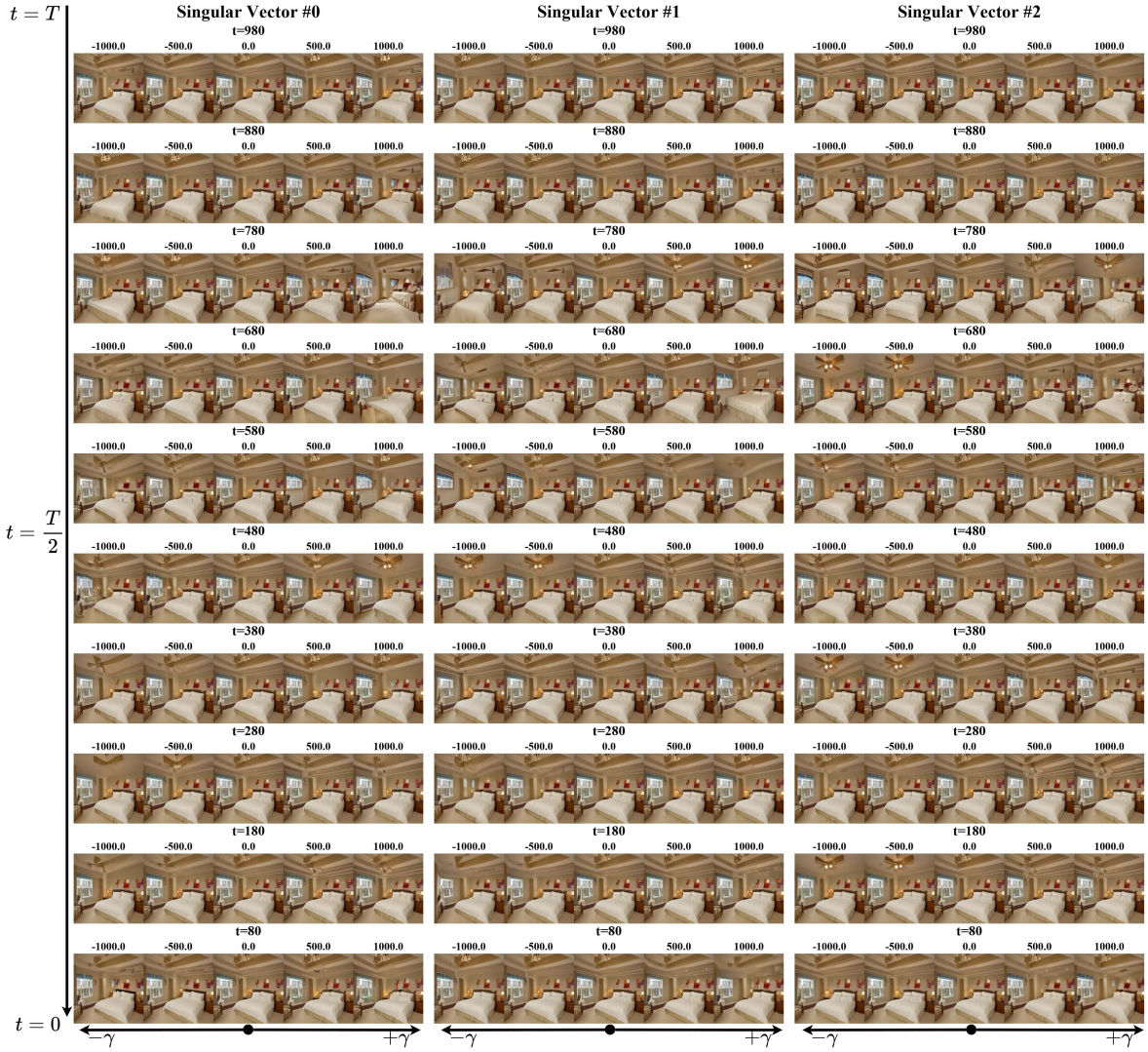


Figure 20. **Directions found with Alg. 1**. For a DDM trained on bedrooms.

Figure 21. **Directions found with Alg. 1**. For a DDM trained on churches.