SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

One-Shot Landmark-Based Face Reenactment

Yoonha Choe

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

One-Shot Landmark-Based Face Reenactment

Gesichtsneugestaltung mit Landmarksbasierter Einzelaufnahme

Author:	Yoonha Choe
Supervisor:	Prof. Dr. Matthias Nießner
Advisor:	Prof. Dr. Matthias Nießner
Submission Date:	15.06.2023

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, 15.06.2023 Yoonha Choe

Acknowledgments

I am deeply grateful to my supervisor, Prof. Dr. Matthias Nießner, for his invaluable guidance and support throughout my master's program. His expertise and encouragement helped me to complete this research and write this thesis.

I would also like to thank my friends and family for their love and support during this process. Without them, this journey would not have been possible.

Finally, I would like to thank all the participants in my study for their time and willingness to share their experiences. This work would not have been possible without their contribution.

Abstract

Face reenactment aims to synthesize realistic images of a source actor with head poses and facial movements synchronized with a specified driving actor. However, current face reenactment methods have several challenges which limit the quality and controllability of the generated videos. In this thesis, we propose a One-Shot Landmark-Based Face Reenactment, which generates reenacted images given a single source identity image conditioned by the facial landmarks of the driving video. Our method generates high-resolution reenacted video by incorporating a pretrained Style-Based Generative Adversarial Network (StyleGAN) generator. We first embed the source identity and driving image in the StyleGAN latent space via optimization. We then train the mapping network, which learns to map the displacement in the facial landmarks of the source and driving image to the displacement in the StyleGAN latent vectors of the source and driving image. We evaluate our model on standard measures and show that our method gives a comparable performance to the recent state-of-the-art methods on the human face dataset. Our source code is publicly available at `https://github.com/yoonhachoe/One-Shot-Landmark-Based-Face-Reenactment`.

Contents

1 Introduction

Face reenactment is an emerging conditional face synthesis task with a given source image or video and a driving/target video. The purpose of face reenactment is to transfer a target's facial movements to a source face while preserving the identity of the source actor. In recent years, face reenactment has attracted enormous research efforts due to its practical value in various applications including virtual reality, video conferencing, movie effects, and entertainment industries [WML21; Nar+20; Cha+19].

Due to the rapid development of deep learning and Generative Adversarial Network (GAN) [Goo+14], many impressive works have been conducted on talking head generation or face reenactment [Pra+20a; Zho+20; Zho+21; Gur+22; WKZ18b; Sia+19a; WML21; Sia+21a; Hon+22a; Gao+23]. Early works of face reenactment require multiple source images to generate one reenactment result, which is considered a few-shot method [Bur+20; DZS21]. Currently, the mainstream works focus on a one-shot generation that uses one single source identity image to generate reenacted results by transferring the facial movement information from the driving frame [WML21; Wan+21; Ji+22; Sia+19a; Zak+20; Yao+21; Zha+19]. Many facial landmark-based face reenactment works utilized self-supervised learned landmarks and dense flow fields to transform the source landmarks and guide the reconstruction of the driving image [Sia+19a; WML21; Sia+21a; Hon+22a]. However, these methods still face many challenges. First, the generated face has unexpected deformation and distortions. Furthermore, no prior framework supports producing a high-resolution output video, *i.e.*, the output resolution is restricted to 256×256 in most cases.

Recent advances in image synthesis have been successful at generating high-resolution images from a latent vector [KLA18; Kar+19; Kar+20; Kar+21a]. Karras *et al*. [KLA18] proposed StyleGAN that synthesizes high-quality images that can be adjusted with "style", and many recent works have studied the latent space of StyleGAN and discovered meaningful semantics for manipulating images [AQW19a; AQW19b; Kar+19; Ric+20a; Tov+21a; Yao+22a]. Inspired by these advances, we propose a novel method for generating high-resolution face reenactment videos using pretrained image generator conditioned on facial landmarks given a single source identity image. Thus, we call our framework One-Shot Landmark-Based Face Reenactment. Our framework consists of two parts: (1) GAN inversion and (2) face reenactment mapping network. We first find the latent code of the given source and driving image in the latent space of the

StyleGAN generator via an optimization process. In this stage, we adopt different types of loss functions and additional regularization to improve both reconstruction quality and the editability of inverted images. We then train the mapping network, which maps the difference between the facial landmarks of the source and the driving frame to the difference between the latent vectors of them. In test time, we add the predicted difference between the latent vectors to the source identity latent vector and get the reenacted image with the identity of the source actor and the facial movements of the target actor. We compare our GAN inversion and face reenactment approaches respectively with the state-of-the-art approaches qualitatively and quantitatively using benchmark measures. The experimental results indicate that our method gives comparable performance to the state-of-the-art methods, generating high fidelity GAN inversion and face reenactment results.

2 Related Work

2.1 Generative Adversarial Network

GAN is a deep generative model that learns to generate new data through adversarial training, considered the groundbreaking work by Goodfellow *et al*. [Goo+14] in 2014. It consists of two neural networks: a generator G, and a discriminator D, which are trained jointly through an adversarial training process. The objective of the generator is to synthesize fake data that resembles real data, while the discriminator tries to distinguish between real and fake data. During the adversarial training process, the generator tries to generate fake data that match the real data distribution to fool the discriminator. The two models are trained together in the following two-player minimax game until the discriminator model is fooled about half the time, meaning the generator model is being able to generate plausible examples:

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

where $p_{\text{data}}(\boldsymbol{x})$ is the distribution of real data, $p_z(\boldsymbol{z})$ is the distribution of generator, $D(\boldsymbol{x})$ is the discriminator network, and $G(\boldsymbol{x})$ is the generator network.

Many GAN models have been developed to produce significantly more realistic images and synthesize them at much higher resolution. For image generation, Deep Convolutional Generative Adversarial Network (DCGAN) is the first milestone that lays down the foundation of GAN architectures as fully Convolutional Neural Network (CNN) [RMC16]. Since then, many methods have been proposed to improve the performance, however, due to the limitation of computational power and shortage of high-quality training data, these works are only tested with low-resolution datasets. Furthermore, the generation of high-resolution images is difficult since a higher resolution makes it easier to tell the generated images apart from training images. To handle this issue, Karras *et al*. [Kar+17] collected the first high-quality human face dataset, High-Quality version of CelebFaces Attributes (CelebA-HQ), and introduced a Progressive Growing Generatie Adversarial Network (PGGAN) for a high-resolution image generation task. The key idea of PGGAN is to grow both the generator and discriminator progressively, starting from low-resolution images and adding new layers that introduce high-resolution fine details as training progresses. This both speeds

up the training process and stabilizes it, allowing the model to generate images of high-quality, such as CelebA-HQ images at a resolution of 1024×1024. However, the generation of high-quality images from complex datasets, *e.g.*, ImageNet, remains an elusive goal. To this end, Brock *et al*. [BDS18] proposed BigGAN, which is trained with two to four times as many parameters and eight times the batch size compared to the prior art. Brock *et al*. [BDS18] introduced general architecture changes that improve scalability, and modified a regularization scheme to improve conditioning, demonstrably boosting performance. They showed that training of GAN benefits dramatically from scaling, and BigGAN can generate realistic synthetic images and smooth interpolations spanning different classes.

2.2 Style-based Generator

In 2018, Karras *et al*. [KLA18] presented a new dataset of human faces called Flickr-Faces HQ (FFHQ) that offers much higher quality and covers wider variation than existing high-resolution datasets. At the same time, they introduced a style-based generator architecture, named StyleGAN, inspired by the idea of neural style transfer [Li+17], which further improves the performance of GAN on human face generation tasks. The following is the overview of the whole family of StyleGAN including the latest version, StyleGAN3.

StyleGAN While a traditional generator [Kar+17] feeds the latent code through the input layer only, StyleGAN first maps the input to an intermediate latent space $\mathcal{W}$, which controls the generator Adaptive Instance Normalization (AdaIN) at each convolution layer. The generator can adjust the "style" of the image at each convolutional layer based on the latent code, therefore controlling the strength of image features at different scales. This new generator architecture leads to disentangled properties of the latent space $\mathcal{W}$, which allow one to perform image manipulations by leveraging a pretrained StyleGAN. The perceptual quality and variety of the StyleGAN synthetic images exceeded the traditional generative models.

StyleGAN2 In StyleGAN2, Karras *et al*. [Kar+19] proposed changes in both model architecture and training methods to further improve the synthetic image quality. StyleGAN2 redesigned the generator normalization, which removes the blob-like artifacts in the images. Additionally, StyleGAN2 analyzed artifacts related to progressive growing, which are called phase artifacts, and proposed an alternative design that achieves the same goal without changing the network topology during training. Both Fréchet Iception Distance (FID) [Heu+17] and precision and recall [Saj+18] are metrics that are used to measure the quality of images. However, they focus on textures rather than shapes, thus, they do not accurately capture all aspects of image quality. StyleGAN2

observed that the Perceptual Path Length (PPL) metric [KLA18], originally introduced as a method for estimating the quality of latent space interpolations, correlates with the consistency of shapes, and it can be used to regularize the synthesis network which leads to an improvement in quality of images.

StyleGAN2-Ada StyleGAN2-Ada [Kar+20] proposed an adaptive discriminator augmentation mechanism that prevents the model from overfitting when training in limited data regimes. This approach does not require any changes to a loss function or network architecture, and it can be applied both when training from scratch or fine-tuning a pretrained GAN. StyleGAN2-Ada trained the discriminator and generator only with an augmented dataset and controlled the augmentation strength dynamically based on the degree of overfitting.

StyleGAN3 StyleGAN2 had a problem with "texture sticking", which does not synthesize images in a natural hierarchical manner, instead, the fine detail appears to be fixed in pixel coordinates. It is caused by careless signal processing that leads to aliasing in the generator network. StyleGAN3 [Kar+21a] proposed an alias-free generator architecture to build the hierarchical synthesis process. To suppress aliasing, StyleGAN3 converts the StyleGAN2 [Kar+19] generator to be fully equivariant to two types of transformation: translation and rotation, and four types of operation: convolution, upsampling, downsampling, and nonlinearity by employing small architectural changes.

2.3 Generative Adversarial Network Inversion

To perform a semantic image editing operation that can be applied to real images, one must first embed a given image into the latent space of the pretrained GAN. This task is referred to as GAN inversion [Xia+21]. The image can then be reconstructed from the embedded code by the generator of pretrained GAN. To successfully invert a real image, one needs to find the latent code that reconstructs the input image accurately and allows for its meaningful manipulation. High-quality inversion is characterized by two aspects: (i) distortion and (ii) editability [Tov+21a]. First, the generator should reconstruct the given image with the inverted latent code, and distortion measures the similarity between the original and inverted images. Second, it should be possible to allow for the subsequent editing of inverted images. Generally, GAN inversion methods either (i) optimize the latent vector to minimize the error for the given image [AQW19a; AQW19b; Kar+19], or (ii) train an encoder to map the given image to the latent code [Tov+21a; Ric+20a; Yao+22a]. The former optimization approach is superior in achieving low distortion. However, it requires a longer time to invert the image, and mostly it is less editable. Many recent works for GAN inversion are changing from

optimization to encoder approach since the downstream editing task is considered the main motivation for inverting an image. Much of the recent literature on GAN inversion pays particular attention to style-based generators [KLA18; Kar+19; Kar+20; Kar+21a] since their latent spaces are better disentangled and have improved editing properties. Here are some popular StyleGAN inversion algorithms in chronological order.

Image2StyleGAN Abdal *et al*. [AQW19a] first proposed a feasible optimization-based algorithm to embed a given image into the latent space of StyleGAN, and there have been many works trying to improve upon this initial idea. Image2StyleGAN (I2S) analyzed deeply that which latent space in StyleGAN [KLA18] should be used for embedding. There are multiple latent spaces in StyleGAN, an initial latent space $\mathcal{Z}$, an intermediate latent space $\mathcal{W}$, and an extended latent space $\mathcal{W}+$ which is a concatenation of 18 different 512-dimensional $\mathbf{w} \in \mathcal{W}$ vectors, one for each layer of the StyleGAN architecture that can receive as input via AdaIN. I2S showed that embedding into $\mathcal{W}+$ gives more reasonable results than embedding into $\mathcal{Z}$ or $\mathcal{W}$. Furthermore, Abdal *et al*. [AQW19a] investigated the initialization of latent code and showed initialization to the mean latent vector $\bar{\mathbf{w}}$ leads to better results than random initialization. To measure the similarity between the input image and embedded image during optimization, they used a weighted combination of the VGG-16 perceptual loss [JAF16] and the pixel-wise MSE loss as a loss function.

Image2StyleGAN++ Abdal *et al*. [AQW19b] introduced the extended version of I2S, which is called Image2StyleGAN++ (I2S++). It improved the reconstruction quality of I2S [AQW19a] by incorporating a noise optimization step to restore the high-frequency details in the input image. I2S++ also extended the global $\mathcal{W}+$ latent space embedding, which allows for local modifications such as missing regions and locally approximate embeddings. The combination of embedding and activation tensor manipulation helps I2S++ to perform high-quality local edits along with global semantic edits on input images.

StyleGAN2 Whereas previous research [AQW19a] suggests finding the latent code in the extended latent space $\mathcal{W}+$, StyleGAN2 [Kar+19] focuses on finding latent codes in the unextended latent space $\mathcal{W}$ since embedding images into the $\mathcal{W}+$ space sacrifices editing quality to achieve better reconstruction and the $\mathcal{W}$ space corresponds to images that the generator could have produced. StyleGAN2 projection method mainly differs from others in optimizing not only the latent code but also the stochastic noise inputs of the generator, regularizing them to ensure they do not end up carrying a coherent signal. Furthermore, it added ramped-down noise to the latent code during the optimization in order to explore the latent space more comprehensively.

Pixel2Style2Pixel Using an optimization-based approach, a fast and accurate inversion of real images into the $\mathcal{W}+$ space remains a challenge. Pixel2Style2Pixel (pSp)

proposed a novel encoder architecture to directly map the given image to a series of style vectors which are fed into a pretrained generator, forming the extended $\mathcal{W}+$ latent space [Ric+20a]. The encoder is based on a Feature Pyramid Network (FPN) [Lin+16], where style codes are extracted from different pyramid scales and inserted into the pretrained StyleGAN generator in correspondence to their spatial scales. Richardson *et al*. [Ric+20a] showed that pSp can directly reconstruct real input images without requiring a time-consuming optimization process. pSp also has the potential for facial image-to-image translation tasks.

Encoder for Editing To successfully invert a real image, Tov *et al*. [Tov+21a] deeply analyzed the existence of a distortion-editability tradeoff within the StyleGAN latent space. Abdal *et al*. [AQW19a] demonstrated that any image can be inverted into an extended latent space $\mathcal{W}+$ since the $\mathcal{W}+$ space has more degrees of freedom and is more expressive than $\mathcal{W}$ space. However, inverting images away from the original $\mathcal{W}$ space reaches regions of the latent space that are less editable and where the perceptual quality is lower. To achieve both reasonable distortion and editability, Tov *et al*. [Tov+21a] proposed two key properties. First, the low variance between the different style vectors. Second, each style vector should lie within the distribution $\mathcal{W}$. Based on these properties, Encoder for Editing (e4e) was introduced, which is specifically designed for facilitating editing on real images by balancing the tradeoff. e4e builds upon the pSp encoder [Ric+20a], but designed specifically for editing. Unlike the original pSp encoder which generates N style vectors in parallel, e4e infers a single style code and a set of offsets and minimizes the variance between the different style codes, which can be seen as a novel progressive training scheme.

Feature-Style Encoder Yao *et al*. [Yao+22a] proposed a novel architecture for GAN inversion, called Featyre-Style Encoder (FSE), which achieves better perceptual quality, lower reconstruction error, and more accurate image editing capacity than existing methods. Yao *et al*. [Yao+22a] pointed out several limitations from previous encoder-based methods: large reconstruction error, lack of fine details, and failure of inversion on outlier data. FSE tackled the weakness by exploiting the idea of encoding feature and style code separately. The feature code encodes spatial details, and the latent code is used for editing. This architecture significantly improves the perceptual quality of the inversion and obtains a balanced trade-off between reconstruction quality and editability.

2.4 Face Reenactment

Face reenactment aims to animate a source image or video using a driving video's facial movements while preserving the source identity. Face reenactment works can be

classified into two categories based on the type of input they use to generate a reenacted video: Audio-driven [Pra+20a; Zho+20; Zho+21; Gur+22] and facial landmark-driven [Sia+19a; WML21; Sia+21a; Hon+22a; Gao+23]. Audio-driven face reenactment methods are capable of generating high-quality lip sync since the audio does not contain identity information, however, they have a drawback in handling non-verbal cues. On the other hand, video-driven methods highly rely on the disentanglement of motion from appearance. We discuss previous works in both audio-based and video-based face reenactment in the following sections.

2.4.1 Audio-driven face reenactment

Wav2Lip Prajwal *et al*. [Pra+20a] proposed Wav2Lip, a novel lip-synchronization network, which is more accurate than previous works for lip-syncing arbitrary talking face videos with arbitrary speech. Previous works can only produce lip movements on the image or videos of specific people seen during the training phase, however, Wav2lip solved this issue to morph the lip movements of arbitrary identities by learning from a strong lip sync discriminator which penalizes incorrect lip shapes. Prajwal *et al*. [Pra+20a] pointed out that the pixel-level $\mathcal{L}_1$ reconstruction loss and the discriminator loss used in the existing works [CJZ17; Pra+20b] are inadequate to penalize inaccurate lip sync generation. Based on these findings, they proposed to use a pretrained expert lip sync discriminator that is accurate in detecting sync in real videos by adapting the modified version of SyncNet [CZ16].

MakeItTalk Zhou *et al*. [Zho+20] presented MakeItTalk, which generates expressive talking head videos from a single facial image with audio as the only input. In contrast to previous methods to learn direct mappings from audio to raw pixels for generating talking faces, MakeItTalk first disentangles the speech content and speaker identity features in the input audio signal. Thus, the audio content can control the motion of lips and nearby facial regions, while the speaker information determines the rest of the talking head dynamics. This disentanglement leads to significantly more plausible face animations. It is able to animate facial landmarks of a given human face or even a non-photorealistic cartoon image in a speaker-aware fashion.

PC-AVS While the previous works achieved reasonable lip synchronization for arbitrary identity, they lack the ability to control head poses since 3D is not involved. To circumvent this problem, Zhou *et al*. proposed [Zho+21] Pose-Controllable Audio-Visual System (PC-AVS), a method that controls the head pose of the talking face by disentangling identity, speech content, and poses. Instead of learning pose motions from audio, they leverage another pose source video to compensate only for head motions. The key is to devise an implicit low dimension pose code that is free of mouth shape or identity information inspired by 3D pose priors in talking faces.

SPACE Gururani *et al*. [Gur+22] presented a method called Speech-driven Portrait Animation with Controllable Expression (SPACE), which generated high-quality resolution animation with control over not only the output pose but also emotions and intensities of expressions. SPACE decomposes the task into several subtasks that allow for better interpretability and fine-grained controllability. While previous approaches [Zho+20; Wan+21; Ji+22] have strategies where the audio is mapped to an intermediate representation such as facial landmarks or latent keypoints, SPACE is the first method that utilized both of them simultaneously as intermediate face representations.

2.4.2 Facial landmark-driven face reenactment

X2Face Wiles *et al*. introduced a network, X2Face [WKZ18b], a novel self-supervised network architecture that can be used for face puppeteer of a source face given a driving vector, *i.e*., a driving frame. The embedding network learns an embedded face representation for the source face, effectively face frontalization, and the driving network learns how to map from this embedded face representation to the generated frame via the driving vector.

FOMM First Order Motion Model (FOMM) was published by Siarohin *et al*. [Sia+19a] in 2019, and it is still considered the most influential work in the facial landmark-driven talking face generation area. The key idea of FOMM is to estimate the motion field from sparse keypoints detected in both source and driving frames, and then combine the appearance extracted from the source image and the motion derived from the driving video. FOMM surpassed the previous work called Monkey-Net [Sia+18] by using a set of self-learned keypoints along with local affine transformations and an occlusion-aware generator that adopts an occlusion mask automatically estimated to indicate object parts that are not visible in the source image and that should be inferred from the context.

OSFV Wang *et al*. [WML21] proposed One-Shot Free-View neural talking head synthesis (OSFV) in 2021. Existing 2D-based one-shot talking head methods [Sia+19a; Wan+19; Zak+20] can only synthesize the talking head from the original viewpoint due to the absence of 3D graphics models. OSFV addressed the fixed viewpoint limitation and achieved local free-view synthesis by using a novel 3D keypoint representation, where person-specific and motion-related information is decomposed. Using the decomposition, 3D transformations can be applied to the person-specific representation to simulate head pose changes, such as rotating the talking head in the output video.

MRAA Siarohin *et al*. [Sia+21a] presented Motion Representations for Articuated Animation (MRAA) with several contributions over FOMM [Sia+19a]. In contrast to the previous keypoint-based works [WKZ18b; Sia+19a], MRAA extracts meaningful and consistent regions instead of keypoints, describing locations, shape, and pose to

generate a warpable motion field. The regions correspond to semantically relevant and distinct object parts, that are more easily detected in frames of the driving video. This enables more stable object and motion representations.

DaGAN Depth-aware GAN (DaGAN) [Hon+22a] was first introduced as a self-supervised geometry learning method to automatically recover the dense 3D geometry from the face videos without the requirement of any expensive 3D annotations. Based on the learned dense depth maps, Hong *et al*. [Hon+22a] further proposed to leverage them to estimate sparse facial keypoints that capture the critical movement of the human head. The depth is also utilized to learn 3D-aware cross-modal (*i.e*., appearance and depth) attention to guide the generation of motion fields for warping source image representations.

PECHead Gao *et al*. proposed [Gao+23] Pose and Expression Controllable Head model (PECHead), which can generate high fidelity video face reenactment results and enable talking head video generation with full control over head pose and expression. The learned landmarks-based approaches [Sia+19a; WML21; ZZ22], which only utilize the 2D learned landmarks without face shape constraints, often produce unexpected deformation and distortions. PECHead addressed the challenge by leveraging head movements to control the estimation of learned and predefined landmarks, enabling free control over the head pose and expression in face reenactment.

3 Method

3.1 Overview

Our proposed approach takes a source identity image and a driving video as input and produces a reenacted output video with the identity of the source image and the facial movement of the driving video. Our method consists of two parts: (1) GAN inversion and (2) face reenactment mapping network. The two parts are described in Section 3.2 and Section 3.3, respectively.

3.2 Generative Adversarial Network Inversion

Before building the face reenactment network, we first need to find the latent vector of the input images. To embed the input image into the StyleGAN latent space, while many recent works for GAN inversion are encoder-based approaches, we choose a simple optimization-based approach since there is no need for effort to choose a training dataset or decoder's architecture design. Figure 3.1 shows the overall optimization process. As we have seen in Section 2.3, the optimization approach has a problem in that the inverted image has low distortion but is less editable than the encoder approach. To balance the distortion-editability tradeoff, we bring the idea of regularization of latent code from e4e [Tov+21a] and have several changes to the loss function from typical optimization-based approaches [AQW19a; AQW19b]. We embed input images into StyleGAN3 [Kar+21a] since it is the latest released version of StyleGAN, and for simplicity and convenience, we refer to StyleGAN3 as simply StyleGAN. Algorithm 1 shows the pseudocode of our GAN inversion optimization approach.

3.2.1 Which Latent Space to Choose?

There are multiple latent spaces in StyleGAN [Kar+21a] which can be used for embedding. StyleGAN consists of a mapping network and the generator G. The mapping network first maps a random 512-dimensional latent code $\mathbf{z} \in \mathcal{Z}$ to an intermediate 512-dimensional latent code $\mathbf{w} \in \mathcal{W}$ via a fully connected neural network. Then, learned affine transformations specialize the intermediate latent code $\mathbf{w}$ to styles that scale and bias the feature maps after each convolution layer of the synthesis network G. Many

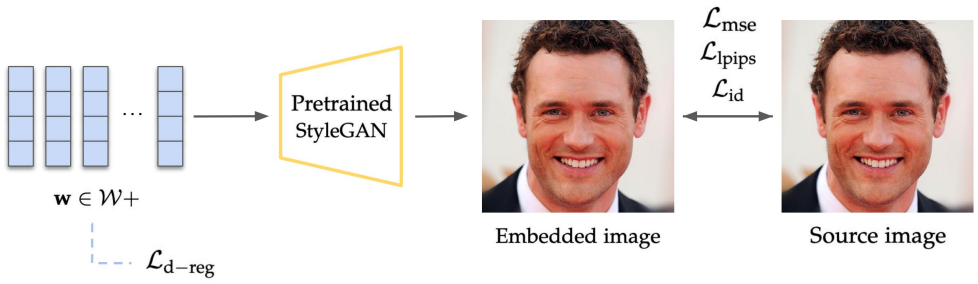

Figure 3.1: Overview of the GAN inversion optimization process. Given the source image to embed and the pretrained StyleGAN, we optimize the latent code $\mathbf{w} \in \mathcal{W}+$ which is a concatenation of 16 different 512-dimensional latent codes with different kinds of losses. During the optimization, $\mathcal{L}_{\text{d-reg}}$ regularization encourages small variance between latent codes, which improves editability. $\mathcal{L}_{\text{mse}}$, $\mathcal{L}_{\text{lpips}}$, and $\mathcal{L}_{\text{id}}$ guide the latent code to generate an image close to the source image.

Algorithm 1: Optimization for Latent Space Embedding

Input : An image $I \in \mathbb{R}^{n \times n \times 3}$; a pretrained generator $G(\cdot)$; gradient-based optimizer F'.

Output: The embedded latent code $\mathbf{w}^*$; the embedded image $G(\mathbf{w}^*)$.

Initialize latent code with the mean latent code $\mathbf{w}^* = \bar{\mathbf{w}}$;

while *not converged* **do**

  $\mathcal{L} \leftarrow \lambda_{\text{lpips}} \frac{1}{N} \|G(\mathbf{w}^*) - I\|_2^2 + \lambda_{\text{lpips}} \sum_{i=0}^{4} \frac{1}{N_i} \|F_i(G(\mathbf{w}^*)) - F_i(I)\|_2^2 + \lambda_{\text{id}}(1 - \langle R(G(\mathbf{w}^*), R(I)\rangle) + \lambda_{\text{d-reg}} \sum_{i=1}^{15} \|\Delta_i\|_2^2$;

  $\mathbf{w}^* \leftarrow \mathbf{w}^* - \eta F'(\nabla_{\mathbf{w}^*} \mathcal{L})$;

end

GAN inversion works analyzed densely that it is difficult to project a real image directly to the original latent spaces $\mathcal{Z}$ or $\mathcal{W}$ due to the gap between the real and synthetic data distribution [AQW19a; Kar+19; Tov+21a; Yao+22a]. Abdal *et al*. [AQW19a] first proposed to embed real images into an extended latent space $\mathcal{W}+$, where $\mathbf{w} \in \mathcal{W}+$ is a concatenation of 16 different 512-dimensional latent codes $(\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{15})$, each one controlling a corresponding convolutional layer in the StyleGAN generator that can receive as a style input. They showed that embedding into $\mathcal{W}+$ offers more expressive power and improves the reconstruction quality than embedding into the original latent spaces. Thus, based on the results of previous works, we adopt the way of embedding images into the extended latent space $\mathcal{W}+$.

3.2.2 Initialization of Latent Code

In the case of optimization-based GAN inversion, the initialization of latent code has a significant impact on the quality of the inverted image. Abdal *et al*. [AQW19a] first investigated two design choices for the initialization of latent code: random initialization and mean latent vector initialization. The second choice is motivated by the observation that the distance to the mean latent vector $\bar{\mathbf{w}}$ can be used to identify low-quality faces [KLA18]. Thus, using $\bar{\mathbf{w}}$ as initialization can lead the optimized vector $\mathbf{w}^*$ to converge to the vector which is close to $\bar{\mathbf{w}}$. Abdal *et al*. [AQW19a] showed that initialization with the mean latent vector $\bar{\mathbf{w}}$ also achieves a much lower reconstruction and perceptual loss value between the input image and the inverted image than random initialization. We adopt the mean latent vector initialization and compute $\bar{\mathbf{w}}$ by averaging 10,000 latent codes $\mathbf{w} \in \mathcal{W}+$ which are computed from random 10,000 latent codes $\mathbf{z}$ through the mapping network from StyleGAN.

3.2.3 Optimization Loss

During the optimization process, one must measure the similarity between the input image and the embedded image. Given the input image $I \in \mathbb{R}^{n \times n \times 3}$ and the pretrained generator G from StyleGAN, we optimize the embedded latent code $\mathbf{w} \in \mathcal{W}+$ which reconstructs the input image minimizing the loss function, thereby decreasing the distortion and increasing the editability of the reconstructed image.

Pixel-wise reconstruction loss To reconstruct the input image, a pixel-wise Mean Squared Error (MSE) is the loss one can intuitively think of to decrease the distortion, which is expressed as:

$$\mathcal{L}_{\text{mse}}(\mathbf{w}) = \frac{1}{N} \| G(\mathbf{w}) - I \|_2^2, \tag{3.1}$$

where N is the number of scalars in the input image, *i.e*., $N = n \times n \times 3$.

Perceptual loss I2S employed a VGG-16 perceptual loss [JAF16] for measuring perceptual similarities between the input image and the embedded image [AQW19a]. The feature distances in the VGG network are computed as cosine distance in the channel dimension and averaged across spatial dimensions and layers of the network, which can be used for measuring the feature similarity between two images. While the features of the VGG network have been useful as optimization loss for image reconstruction, Zhang *et al*. [Zha+18] analyzed how different perceptual losses actually correspond to human visual perception and how they compare to traditional perceptual image evaluation metrics, such as $\mathcal{L}_2$ Euclidean distance, Structural Similarity Index Measure (SSIM) [Wan+04], Mean Structural Similarity Index Measure (MSSIM) [WSB03] or Feature Similarity (FSIM) [Zha+11]. They showed that all types of deep features including VGG network outperform the traditional perceptual metrics and found that AlexNet [KSH12] is the best for perceptual similarity. This new perceptual loss is referred to as a Learned Perceptual Image Patch Similarity (LPIPS) loss [Zha+18]. Based on the results of prior work, we employ AlexNet [KSH12] perceptual loss rather than VGG-16 perceptual loss in addition to the pixel-wise MSE loss. Our LPIPS loss is defined as:

$$\mathcal{L}_{\text{lpips}}(\mathbf{w}) = \sum_{i=0}^{4} \frac{1}{N_i} \|F_i(G(\mathbf{w})) - F_i(I)\|_2^2 , \tag{3.2}$$

where F_i is the feature output of AlexNet `conv1-conv5` layers and N_i is the number of scalars in the i^{th} layer output.

Identity loss A common challenge when handling the task of encoding facial images is the preservation of the input identity. To tackle this, we employ an Additive Angular Margin Loss (ArcFace), which is a loss function used in face recognition tasks [DGZ18]. Face recognition is the task of making a positive identification of a face in an image against a pre-existing database of faces, and the loss function for this task is often called identification loss, ID loss in short. The softmax is traditionally used in these tasks, however, it does not explicitly optimize the feature embedding to enforce higher similarity for intraclass samples and diversity for interclass samples. To handle this issue, ArcFace is proposed to stabilize the training process and further improve the discriminative power of the face recognition model. pSp and e4e [Ric+20a; Tov+21a] also adopted ArcFace ID loss during the evaluation, measuring the cosine similarity between the input and embedded images. This term is defined as:

$$\mathcal{L}_{\text{id}}(\mathbf{w}) = 1 - \langle R(G(\mathbf{w}), R(I) \rangle , \tag{3.3}$$

where R is the pretrained ArcFace network.

Delta-regularization e4e [Tov+21a] first pointed out that if one optimizes the latent code in the extended latent space $\mathcal{W}+$, the embedded latent codes can be far away

from the original trained latent space $\mathcal{W}$ during the optimization, which makes the inverted images have low editability. To encourage the optimized latent code to converge into regions close to $\mathcal{W}$, we bring the idea from e4e, which minimizes the variance between the different 16 style codes. The embedded latent code $\mathbf{w}$ is a concatenation of 16 different 512-dimensional latent codes $(\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{15})$, and we can rewrite $\mathbf{w} = (\mathbf{w_0}, \mathbf{w_0} + \Delta_1, \ldots, \mathbf{w_0} + \Delta_{15})$ with a single style code $\mathbf{w}_0$ and a set of offsets $\Delta_1, \ldots, \Delta_{15}$. In this thesis, to explicitly enforce proximity to $\mathcal{W}$, we add an $\mathcal{L}_2$ delta-regularization loss:

$$\mathcal{L}_{\text{d}-\text{reg}}(\mathbf{w}) = \sum_{i=1}^{15} \|\Delta_i\|_2^2 . \tag{3.4}$$

Total loss Bringing all the above losses together, our overall loss object for the optimization of the latent embedding is defined as a weighted combination of the losses,

$$\mathcal{L}(\mathbf{w}) = \lambda_{\text{mse}}\mathcal{L}_{\text{mse}}(\mathbf{w}) + \lambda_{\text{lpips}}\mathcal{L}_{\text{lpips}}(\mathbf{w}) + \lambda_{\text{id}}\mathcal{L}_{\text{id}}(\mathbf{w}) + \lambda_{\text{d}-\text{reg}}\mathcal{L}_{\text{d}-\text{reg}}(\mathbf{w}) , \tag{3.5}$$

where $\lambda_{\text{mse}} = 0.1$, $\lambda_{\text{lpips}} = 1.5$, $\lambda_{\text{id}} = 0.05$, and $\lambda_{\text{d}-\text{reg}} = 0.002$ are empirically obtained weights balancing each loss for good performance.

3.3 Face Reenactment

The goal of our face reenactment method is to synthesize a video of the source identity with the pose and facial movement of the actor from the driving video. An overview of our entire method is shown in Figure 3.2. The training pipeline is as follows. We take the source identity image and the driving frame from the driving video whose identity is the same as the identity of the source image. We first extract facial landmarks from the two input images using a pretrained landmark detector and find the StyleGAN latent vector of the two input images using our GAN inversion method. We then train our face reenactment mapping network, which takes the source and driving facial landmark images and predicts the displacement in the latent vector of the source and driving images. During the inference time, the identity of the source and the driving image can be different. We first extract facial landmarks from the two input images and find the latent vector of only source images. Then, we can get the reenacted latent vector by adding the predicted latent vector from the mapping network to the source latent vector. In the following section, we describe the architecture of our approach in detail.

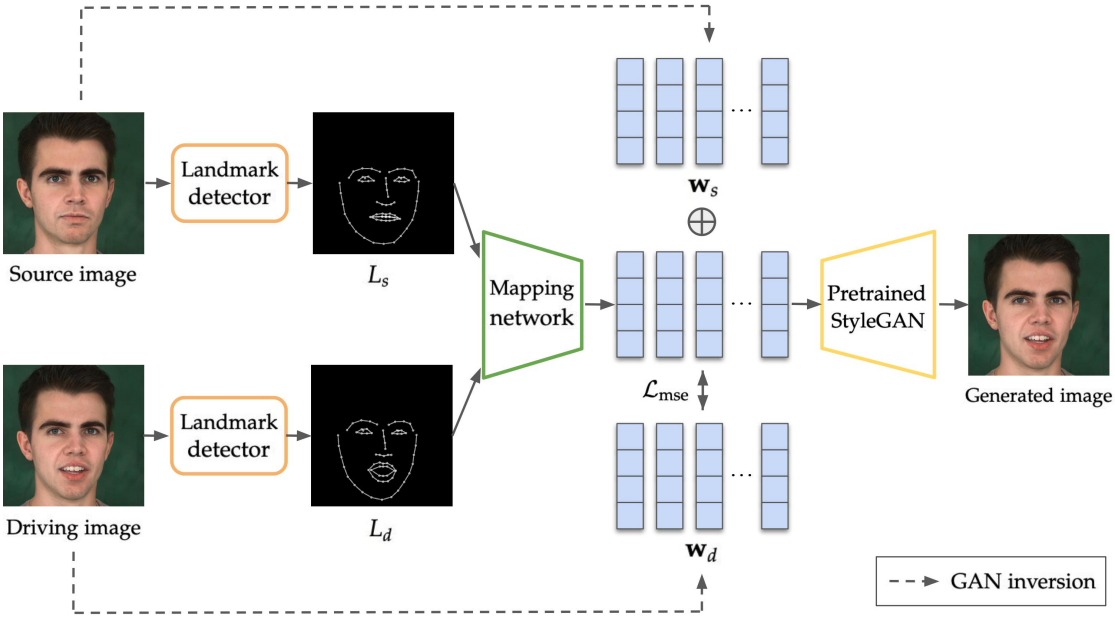

Figure 3.2: Overview of our face reenactment method. Given a source identity image and a frame from the driving video, the aim is to synthesize a video of the source identity with the facial movement of the driving video. We first extract facial landmarks from the source and driving images using a landmark detector, OpenPose [Cao+19]. Next, the mapping network learns to map the difference between landmarks, $L_d - L_s$, to the difference between latent vectors of the source and the driving image, $\mathbf{w}_d - \mathbf{w}_s$. It can be done by minimizing $\mathcal{L}_{\text{mse}}$ between the sum of the predicted latent vector displacement and the source identity latent vector, and the driving latent vector during the training. Lastly, the reenacted image is generated by passing the predicted latent vector to the pretrained StyleGAN.

3.3.1 Network Architecture

Landmark detector To extract the facial landmarks from the source and driving images, we use OpenPose [Cao+19], which is a real-time human pose detection library that has for the first time shown the capability to jointly detect the human body, foot, hand, and facial keypoints on single images. We extract keypoints from the source and driving images using a face detector [Sim+17] from OpenPose [Cao+19] which detects 70 facial keypoints from the face in the input image. We use facial keypoints information as an image for the input of the mapping network, L_d and L_s, respectively, with the size of $3 \times H \times W$, where H and W are the height and width of the input image as shown in Figure 3.2. We resize the facial landmark images to 580×580 before feeding them into the landmark detector.

Mapping network Our mapping network takes two inputs: the source facial landmark image and the target facial landmark image. Our mapping network has a CNN architecture which consists of a set of convolutional blocks and fully connected layers. We build our mapping network with 8 convolutional blocks with a pooling layer followed by 2 fully connected layers. The mapping network is the only network that is trained in our method, and the number of trainable parameters of the mapping network is about 19M. The mapping network computes the displacement in the two facial landmark images, $L_d - L_s$, and through the convolutional blocks and fully connected layers, it learns to predict the displacement in the two latent vectors, $\mathbf{w}_d - \mathbf{w}_s$, which are obtained by GAN inversion. In this way, we kind of remove the identity information by taking the source and driving facial landmark images, and add the identity information by adding the source identity latent vector to the predicted displacement latent vector. The mapping network can learn to map the displacement in the facial landmark image space to the StyleGAN latent space regardless of the driving actor's identity by training multiple different kinds of identities.

3.3.2 Training Loss

Different from other face reenactment methods, our method deal with the source and driving images as the latent vector, not the image. For the training loss function, we calculate a MSE loss between the ground truth driving latent code $\mathbf{w}_d$ and the sum of the predicted latent displacement $\mathbf{w}$ and the source identity latent code $\mathbf{w}_s$. We define $\mathcal{L}_{\text{mse}}$ as follows:

$$\mathcal{L}_{\text{mse}}(\mathbf{w}) = \frac{1}{N} \|\mathbf{w}_d - (\mathbf{w} + \mathbf{w}_s)\|_2^2, \tag{3.6}$$

where N is the dimension of the latent space $\mathcal{W}+$, *i.e.*, $N = 16 \times 512$. MSE loss between latent vectors is the only loss function for training the mapping network, thus, our

overall loss is defined as:

$$\mathcal{L}(\mathbf{w}) = \lambda_{\text{mse}} \mathcal{L}_{\text{mse}}(\mathbf{w}) \,, \tag{3.7}$$

where $\lambda_{\text{mse}} = 10$ is an empirically obtained weight to get a larger gradient since the loss value is too small.

4 Experiments

We now discuss the datasets, metrics, and experiments used to provide a comprehensive set of evaluations to measure the performance of our proposed GAN inversion and face reenactment methods in Section 4.1 and Section 4.2, respectively. We compare with the state-of-the-art works both qualitatively and quantitatively, as well as ablate our contributions.

4.1 Generative Adversarial Network Inversion

4.1.1 Implementation Details

We perform our experiments using PyTorch [Pas+19]. For StyleGAN, we use the official implementation of StyleGAN3 [Kar+21b]. We use the StyleGAN model pretrained on FFHQ dataset with the resolution of 1024×1024, which is translation and rotation equivariant. During the optimization process, we use the Adam optimizer [KB17] with a learning rate of 0.01, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e^{-8}$. We use 1,000 gradient descent steps for the optimization, taking 15 minutes per image on an 11GB NVIDIA GeForce GTX 1080 Ti GPU. To justify our choice of 1,000 optimization steps, we investigated the change in the total loss value as a function of the number of iterations. As Figure 4.1 shows, the average loss value of the human facial images in CelebA-HQ dataset drops the quickest and converges at around 1,000 optimization steps. Based on this observation, we choose to optimize the latent vector for 1,000 steps in all our experiments.

4.1.2 Dataset and Metrics

Dataset We test our GAN inversion method on CelebA-HQ [Kar+17], which is a large-scale face image dataset containing high-quality 30,000 celebrity images at 1024×1024 resolution. We use the first 1,000 images in CelebA-HQ dataset as evaluation data.
Metrics To properly evaluate our proposed method from the inversion quality aspect, we evaluate the performance of reconstruction quality, perceptual quality, and identity similarity. We adopt the evaluation metrics proposed in FSE [Yao+22a]:

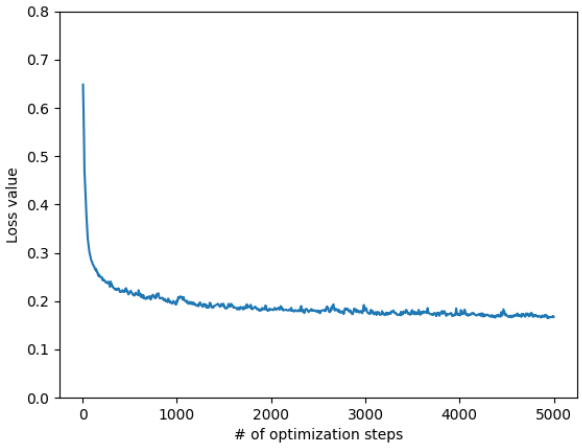

Figure 4.1: Total loss value changes along the optimization steps.

- MSE: MSE is used to evaluate the reconstruction quality of the embedded image compared to the ground truth image. MSE represents the cumulative squared error between the ground truth and the embedded image.
- SSIM [Wan+04]: SSIM evaluates the perceived changes in structural information of an image. It can also handle global illumination changes.
- Peak Signal-to-Noise Raito (PSNR): PSNR also evaluates the reconstruction quality of the embedded image compared to the ground truth image. PSNR represents a measure of the peak error.
- LPIPS [Zha+18]: LPIPS measures perceptual similarities between the ground truth and the embedded image. We compute the cosine similarity between feature distances in AlexNet [KSH12].
- ID loss: We measure the identification similarity between the ground truth and embedded image, which refers to the cosine similarity between the features in ArcFace [DGZ18] of the two images.
- FID [Heu+17]: FID is used to compare the distribution of embedded images with the distribution of ground truth images using the features extracted from an Inception V3 model [Sze+15].

4.1.3 Comparison with State-of-the-art Methods

We evaluate our GAN inversion approach against the state-of-the-art optimization-based method, I2S [AQW19a], and encoder-based methods, pSp [Ric+20a], e4e [Tov+21a], and FSE [Yao+22a]. We use the official implementation [AQW19c; Ric+20b; Tov+21b; Yao+22b] for each method to generate the results. For I2S, we optimized the images for 1,000 steps since Abdal *et al*. [AQW19a] showed that the loss value of human face images converges at around 1,000 optimization steps when using I2S.

For each method, we embed each image in the evaluation dataset to the latent space and Figure 4.2 shows the inversion results of the different methods. We can see that the optimization-based method, I2S [AQW19a], cannot accurately reconstruct the source images. The reconstructed images are not clear, especially fine details such as eyes and teeth in the zoomed-patches have many artifacts. For the encoder-based methods, e4e [Tov+21a], pSp [Ric+20a], and FSE [Yao+22a], there are no artifacts in the embedded images. However, the fine details in the embedded images are slightly different from the source image. As we have seen in Section 2.3, GAN inversion has the distortion-editability tradeoff, and the optimization-based approach generally has difficulty balancing it since the latent vector of the inverted image can be far away from the original latent space during the optimization process. Thus, the optimization-based method tends to have low distortion, trying to reconstruct the image as the same as the input image, however, it makes the reconstructed image not clear and blurry, which also leads to low editability. On the other hand, encoder-based methods sacrifice the distortion quality a bit, *i.e.*, eyes and mouth seem different from the ones in the source image, but generate very clear images, which means high editability. Our method generates quite sharper images than I2S which is the same optimization-based approach but also preserves better the fine details than encoder-based methods. Thus, we can say our method balances the tradeoff well between the clearness of the image and the preservation of fine details. The deeper study of editability is discussed as an ablation study in Section 4.1.4.

Quantitative GAN inversion results of all the methods are reported in Table 4.1. Compared to other methods, our approach achieves the lowest identity loss, which means the highest identity similarity between the source image and the embedded image. It happens because we adopt ID loss during the optimization process. In terms of identity similarity, our improvement can attain 30%. We can see from LPIPS value that our approach also achieves reasonable perceptual quality. There are no remarkable differences between methods except FSE [Yao+22a] in the performance of distortion quality, *i.e.*, MSE, SSIM, and PSNR metrics.

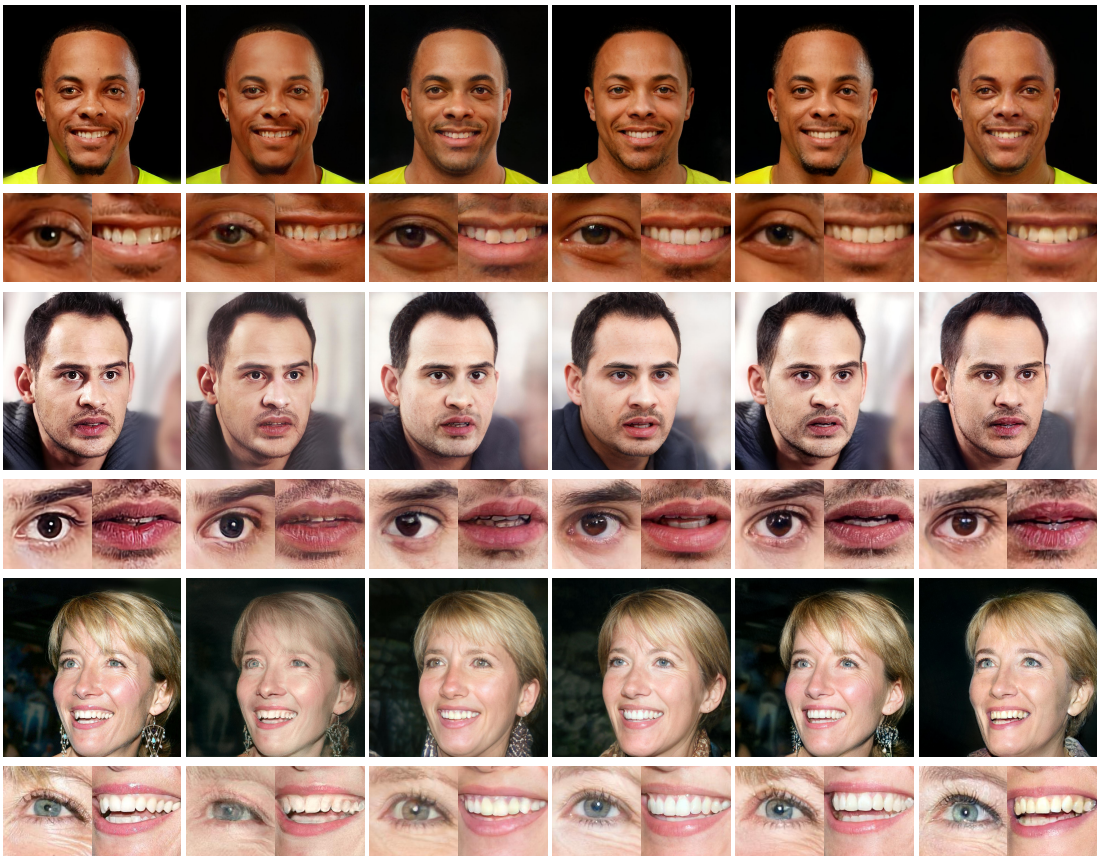

Figure 4.2: Qualitative comparison with the state-of-the-art method for GAN inversion on CelebA-HQ dataset. For details, zoomed-in eye and mouth image patches are provided for each input image.

Method	MSE ↓	SSIM ↑	PSNR ↑	LPIPS ↓	ID ↓	FID ↓
I2S [AQW19a]	0.0111	0.6668	28.7689	0.2035	0.1476	26.5206
pSp [Ric+20a]	0.0103	0.6432	29.2807	0.2771	0.1958	34.9150
e4e [Tov+21a]	0.0137	0.6256	29.1159	0.2997	0.2673	37.1348
FSE [Yao+22a]	**0.0048**	**0.6870**	**30.5188**	**0.1532**	0.0937	**19.9567**
Ours	0.0115	0.6449	29.1421	0.1766	**0.0640**	32.3636

Table 4.1: Quantitative comparison with the state-of-the-art methods for GAN inversion on CelebA-HQ dataset. ↑ indicates larger is better, and ↓ indicates smaller is better.

4.1.4 Ablation Study

We conduct an ablation study on the experimental setup for the inversion of StyleGAN. For setting a baseline that is similar to I2S [AQW19a], we replace AlexNet perceptual loss with VGG-16 perceptual loss and remove identity loss and delta-regularization. We change one experimental setup at a time to this baseline and optimize the image in the evaluation dataset: (A) replace VGG-16 with AlexNet for perceptual loss, (B) add identity loss, and (C) add delta-regularization.

Figure 4.3 shows the qualitative results of the ablation study on the experimental setup for the GAN inversion. Overall, visual inspection shows that our full method outperforms other methods. The faces generated from the full method are the most faithfully reconstructed, while other methods have difficulties preserving fine details such as mustaches, hairstyles, or teeth, and even global characteristics such as head poses. Replacing with AlexNet has the biggest visual improvement from the baseline results among other methods, except for the full method, but still generates images with noticeable artifacts.

We compare the quantitative metrics of several ablative configurations and report them in Table 4.2. For (A), we observe a comparable result on the perceptual quality metric, LPIPS, which means using AlexNet rather than VGG-16 model improves the perceptual quality of the inverted images. Replacing with AlexNet also slightly improves identity similarity between the source and the inverted images. (B) confirms that including ID loss helps to generate the inversion with the same identity, and the improvement of identity similarity is above 90%. These demonstrate that using AlexNet for perceptual loss and adding ID loss in the optimization helps to improve perceptual quality and identity similarity of the inversion results, respectively. For (C), adding delta-regularization embeds images with lower quality than the baseline in

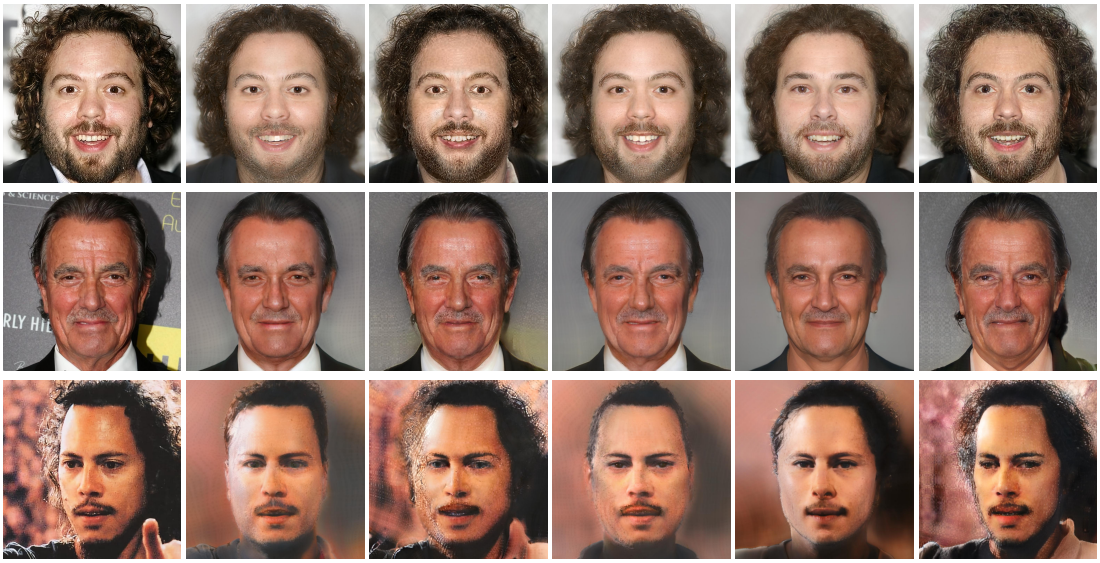

Figure 4.3: Qualitative results of ablation study on experimental setup for GAN inversion. We conduct experiments on three different configurations: (A) replacement VGG-16 with AlexNet for perceptual loss, (B) baseline with ID loss, and (C) baseline with delta-regularization.

Configuration	MSE ↓	SSIM ↑	PSNR ↑	LPIPS ↓	ID ↓	FID ↓
Baseline	0.0147	0.6791	29.1003	0.2406	0.3716	38.2854
+ AlexNet perceptual loss	**0.0095**	0.6678	**29.6151**	**0.1351**	0.2813	36.2275
+ ID loss	0.0144	**0.6831**	29.1560	0.2398	**0.0341**	33.8647
+ Delta-reg	0.0181	0.6600	28.7593	0.2648	0.5343	32.6950
Full	0.0115	0.6449	29.1421	0.1766	0.0640	**32.3636**

Table 4.2: Quantitative results of ablation study on experimental setup for GAN inversion. The baseline represents the model with VGG-16 perceptual loss and without ID loss, and delta-regularization. We conduct experiments on three different configurations: (A) Replacement VGG-16 with AlexNet for perceptual loss, (B) Baseline with ID loss, and (C) Baseline with delta-regularization. ↑ indicates larger is better, and ↓ indicates smaller is better.

terms of reconstruction and perceptual quality, as well as identity similarity. Adding delta-regularization is meant for improving editability, and none of the metrics in Table 4.2 can be used for evaluating the editing capability. Thus, we conduct further ablation study to test if adding delta-regularization loss improves the editability of the reconstructed image.

To evaluate qualitatively the editing results, we project each image in the evaluation dataset to the latent space with and without delta-regularization and apply StyleCLIP [Pat+21b] to generate the editing result on the following facial attributes: expression, beard, and hair color. We use editing via the latent vector optimization method in the official implementation of StyleCLIP [Pat+21a] with the same hyperparameters suggested. The text inputs necessary for editing for each attribute are the following: *"A man with a smiling face"*, *"A man with a beard"*, and *"A man with blonde hair"*. Figure 4.4 shows facial attribute editing results for the method with delta-regularization and without delta-regularization. The inverted images without delta-regularization are closer to the source images than the ones with delta-regularization, *i.e.*, the hand is missing and the fine details in the face are not preserved with delta-regularization. However, when it comes to editing, edited images without delta-regularization have severe artifacts, or changes are small to discern. On the other hand, the inversion method with delta-regularization yields plausible editing results, while at the same time preserving better the sharpness of the image. This result supports that our approach of adapting delta-regularization to the loss function during the optimization process encourages the optimized latent vector close to the StyleGAN latent space and further improves the editability of the embedded image. Both Figure 4.4 and Table 4.2

Figure 4.4: Ablation study on latent space editing. We apply StyleCLIP [Pat+21b] to perform latent editing for facial attribute manipulation for our GAN inversion method with delta-regularization and without delta-regularization.

show that our full method sacrifices the reconstruction and perceptual quality of the inverted images with a small amount that can be ignored and achieves higher identity similarity and editing capability by adapting delta-regularization to the loss function which is motivated from encoder-based inversion methods.

4.2 Face Reenactment

4.2.1 Implementation Details

For the pretrained StyleGAN model, we use the same one as used in GAN inversion. During the training process, we use the Adam optimizer [KB17] with a learning rate of $2e^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e^{-4}$. We use the batch size of 32 and train 20 epochs, taking 1 day and 18 hours on an 11GB NVIDIA GeForce GTX 1080 Ti GPU to train our training dataset.

4.2.2 Dataset and Metrics

Dataset We train and test our face reenactment method on Multi-view Emotional Audio-visual Dataset (MEAD) [Wan+20], which is a talking-face video corpus featuring 60 actors and actresses talking with eight different emotions at three different intensity levels. High-quality audio-visual clips are captured at seven different view angles in a strictly-controlled environment. We build our dataset by choosing neutral emotional

and frontal face videos. We resample those videos to 25 fps and extract every frame for all videos. To align the video frames, we adopt the face alignment algorithm proposed by Bulat *et al*. [BT17]. All frames are cropped based on the face alignment and resized with the size of 1024×1024. We apportion the data into training and test sets, with a 90-10 split. After that, the number of training frames is 181,836 and the number of test frames is 14,938. For training, we first conduct GAN inversion to all frames in our dataset to find each corresponding StyleGAN latent vector. We then have the pair of the input image and the latent vector for all frames. Next, we pick the representative identity image for every actor, which is the frontal and closed-mouth image. We then train the mapping network with the pair of one source identity image and one image from the same actor as a driving image.

Metrics To provide an extensive evaluation of our face reenactment approach, we use several metrics to measure the performance of different works. In self-reenactment experiments, we adopt the evaluation metrics proposed in Agarwal *et al*. [Aga+22], which are $\mathcal{L}_1$, SSIM [Wan+04], PSNR, Landmark Distance (LMD) [Aga+22], and FID [Heu+17]. In cross-identity reenactment experiments, we adopt the evaluation metrics proposed in PECHead [Gao+23], which are ID loss with ArcFace [DGZ18], Average Rotation Distance (ARD) [DZS21], and Action Unit Hamming distance (AUH) [DZS21]. Here are the explanations of the evaluation metrics used in measuring the performance of face reenactment, except the ones that are already explained in Section 4.1.2:

- $\mathcal{L}_1$: $\mathcal{L}_1$ distance is the mean absolute difference between the ground truth frame and the generated frame.
- LMD [Aga+22]: LMD calculates the distance between detected facial landmarks of the ground truth and the generated frame using a pretrained facial landmark detector [KS14]. This metric was denoted by Average Keypoint Distance (AKD) in FOMM [Sia+19a]. In this thesis, we rename it LMD to avoid confusion with the facial landmark detector module used in their work.
- ARD [DZS21]: ARD measures errors of head pose angles between the ground truth and the generated frame.
- AUH [DZS21]: AUH measures errors of facial expressions between the ground truth and the generated frame.

4.2.3 Comparison with State-of-the-art Methods

We evaluate our proposed face reenactment approach against the state-of-the-art facial landmark-based face reenactment methods, X2Face [WKZ18b], FOMM [Sia+19a],

Figure 4.5: Qualitative comparison with the state-of-the-art methods for self-reenactment on MEAD dataset. The source and driving frames are from the same individual.

MRAA [Sia+21a], and DaGAN [Hon+22a]. We use the official implementation [WKZ18a; Sia+19b; Sia+21b; Hon+22b] for all methods to generate the reenacted results.

Self-Reenactment We first compare our models with the above state-of-the-art techniques for self-reenactment, where the source and driving frames are from the same individual. Qualitative results for self-reenactment are shown in Figure 4.5. Figure 4.5 shows that X2Face [WKZ18b] and MRAA [Sia+21a] generate the reenacted image with the severe artifacts, *e.g.*, the head shape is distorted, or the eyes are closed even if the ones from the driving image are not. The generated results of the other two methods, FOMM [Sia+19a] and DaGAN [Hon+22a], are not properly reenacted with the facial landmarks from the driving frame. They struggle in producing a synced mouth with the driving frame. On the contrary, our method generates the images with both a well-preserved source identity and the facial movements from the driving image. Furthermore, other state-of-the-art methods generate only 256×256 resolution at most, while our method generates much higher-resolution images, 1024×1024 due to the pretrained StyleGAN.

Method	$\mathcal{L}_1 \downarrow$	SSIM $\uparrow$	PSNR $\uparrow$	LMD $\downarrow$	FID $\downarrow$
X2Face [WKZ18b]	0.0490	0.5460	30.2264	4.5331	146.5701
FOMM [Sia+19a]	0.0403	0.7783	30.8781	4.5673	125.4573
MRAA [Sia+21a]	**0.0382**	0.8024	**30.9817**	3.9182	**111.9126**
DaGAN [Hon+22a]	0.0425	0.7775	30.5575	4.8296	120.6387
Ours	0.0418	**0.8368**	30.0414	**2.8277**	121.2479

Table 4.3: Quantitative comparison with the state-of-the-art methods for self-reenactment on MEAD dataset. $\uparrow$ indicates larger is better, and $\downarrow$ indicates smaller is better.

Quantitative results for self-reenactment are presented in Table 4.3. For self-reenactment experiments, we compute $\mathcal{L}_1$, SSIM, PSNR, LMD, and FID between the ground truth driving frame and the generated image. Compared to other methods, our approach achieves the highest SSIM value, which means the highest structural similarity between the driving image and the generated image, and the lowest LMD value, which means the most well-reenacted in terms of facial movement. In terms of LMD, our improvement can attain 27%. In terms of $\mathcal{L}_1$ and PSNR value, our method first conducts GAN inversion for the source images, which leads to a lower reconstruction similarity between the ground truth and the generated image. Even considering this weakness, our method achieves reasonable $\mathcal{L}_1$ and PSNR value compared to other state-of-the-art methods.

Cross-identity reenactment Next, we compare our methods with the state-of-the-art methods for cross-identity reenactment, where the source and driving frames depict different individuals. Figure 4.6 shows the qualitative results of cross-identity reenactment experiments. Similar to self-reenactment experiments, X2Face [WKZ18b] and MRAA [Sia+21a] still struggle to produce convincing results with having noticeable face distortion. For FOMM [Sia+19a] and DaGAN [Hon+22a], the identity of the generated images is well-preserved, however, especially the mouth movements are rarely captured from the driving images. In contrast, our method can generate high-quality images with correct mouth and also pupil movement synced with the driving images.

Quantitative results for cross-identity reenactment are shown in Table 4.4. For cross-identity reenactment experiments, we compute ID similarity between the ground truth source image and the generated image, and ARD and AUH between the ground truth driving image and the generated image. Table 4.4 shows that our method outperforms other techniques with the highest identity preservation ability, as well as the lowest

Figure 4.6: Qualitative comparison with the state-of-the-art methods for cross-identity reenactment on MEAD dataset. The source and driving frames are from different individuals.

Method	ID ↓	ARD ↓	AUH ↓
X2Face [WKZ18b]	0.3861	3.0224	0.2256
FOMM [Sia+19a]	0.1142	2.7688	0.1745
MRAA [Sia+21a]	0.2093	2.9065	0.2091
DaGAN [Hon+22a]	0.1077	2.6530	0.1541
Ours	**0.0908**	**1.4558**	**0.1183**

Table 4.4: Quantitative comparison with the state-of-the-art methods for cross-identity reenactment on MEAD dataset. ↑ indicates larger is better, and ↓ indicates smaller is better.

pose angle and facial expression error. The improvement of the identity preservation ability, the head pose angle error, and the facial expression error are 15%, 45%, and 23%, respectively.

5 Discussion

In this section, we state the limitations of our method and discuss future works and the possible impacts of our method on future research.

5.1 Generative Adversarial Network Inversion

Limitations Although our suggested GAN inversion method achieves compelling results, it has several limitations that should be considered. Figure 5.1 shows major challenging cases for our GAN inversion method: the reconstructed image has severe artifacts if (A) the person in the source image is not frontal, (B, C) the person in the source image is wearing a hat or glasses, (D) the facial expression of the person is exaggerated, or (E) the background image has complex patterns. Thus, our model has limitations in the embedded image quality depending on the person in the input image. These cases may be challenging since such examples were not available when training the StyleGAN model. Our GAN inversion method simply optimizes the latent vector in StyleGAN space, while other state-of-the-art methods utilize the complex training scheme, such as learning styles from coarse to fine details progressively [Ric+20a] or learning additional feature code which replaces the pretrained StyleGAN's feature map [Yao+22a]. These techniques prevent the inverted images from having severe artifacts in the abovementioned cases.

Our GAN inversion method can be directly used for other domains that are pretrained with StyleGAN, *e.g.*, MetFaces [Kar+20], which is an image dataset of human faces extracted from works of art, or Animal Faces-HQ (AFHQ) [Cho+19], which is a dataset of animal faces, by simply changing the pretrained model. On the other hand, for the encoder-based method, one needs to train the encoder network whenever using different domains of the images. However, even if our optimization-based method does not require any time-consuming training such as building a training scheme or preparing a training dataset, once the encoder-based method is well-trained, it has much more advantages than the optimization-based method. While our GAN inversion method takes about 15 minutes on an 11GB NVIDIA GeForce GTX 1080 Ti GPU to optimize one single image, encoder-based methods only take less than 100 milliseconds. Thus, if we have a bunch of images to be embedded, the gap in time efficiency between ours and the encoder-based method grows tremendously.

Figure 5.1: Challenging cases for GAN inversion. (A) If the face is not frontal, the side face in the embedded image gets blurry. (B, C) The hat or glasses in the image are not properly captured in the embedded image. They are reconstructed as hair and eyes, respectively, which leads to bad quality. (D) If the facial expression is exaggerated, *e.g.*, the mouth opens too much, the fine details are lost. (E) The background with complex patterns is not preserved in the reconstructed image, which leads to a low MSE value between the source image and the inverted image.

Future Works Editability is an important aspect in GAN inversion since achieving real image editing using the latent space of a pretrained GAN model is considered the main motivation for inverting the image. In this thesis, we could not provide an extensive comparison with the state-of-the-art methods for editability in a quantitative way due to the limitation of time. Thus, the comparison of real image editing capability via different kinds of latent space manipulation [She+20; SZ21] can be done as future work.

5.2 Face Reenactment

Limitations Since our entire method consists of two parts that cannot be trained at one time: (1) GAN inversion and (2) face reenactment, one must conduct GAN inversion to find the StyleGAN latent vector via optimization, which degrades the simplicity of our method. If one wants to train the face reenactment mapping network with a different dataset, one needs to optimize all images in the new dataset via a time-consuming optimization process.

As we discussed in Section 5.1, our GAN inversion approach performs badly if the face in the image is not frontal. That's why we train our face reenactment network with only frontal source and driving images. However, in face reenactment, it is important to generate images regardless of the view angles of the source or driving images. However, we have no clue how our network performs on facial images with different view angles. Our face reenactment network has limitations in controlling head poses and emotional expressions.

Future Works Since our method predicts the StyleGAN latent vector as an output, editing facial attributes such as facial expressions or hair color can be done on the reenacted images. Adopting the StyleGAN latent vector manipulation method, *e.g.*, StyleCLIP [Pat+21b], we can edit the facial attributes of the reenacted images simply with the text inputs as shown in Section 4.1.4, which allows for more freely controllable facial expression.

We use the facial landmark images of the source and the driving frame as inputs for the face reenactment mapping network. As one of the future works, we can study the performance depending on different types of input, *e.g.*, audio that contains emotional expressions, facial landmark coordinates, or multi-modality, and compare the results with our method. Furthermore, our current mapping network has a simple CNN architecture that maps the facial landmark displacement to the latent vector displacement, and we can study adopting more complex architecture such as Long Short-Term Memory (LSTM) using the consecutive inputs from the driving video for the mapping network as future work.

For the training loss, we only use MSE loss between the driving latent vector and the predicted latent vector. Intuitively thinking, our training scheme also can have MSE or LPIPS loss between the driving image and the generated image using pretrained StyleGAN. However, we have limited usage of GPU, and loading the pretrained StyleGAN model during the training leads to out-of-memory, we could not adopt a loss function between the images. If there is enough space for GPU, there is still room for improvement in the performance of face reenactment by using a more complex loss function.

6 Conclusion

In this thesis, we presented a novel approach for generating high-resolution face reenactment videos using pretrained image generator StyleGAN conditioned on facial landmarks given a single source identity image and a driving video. To embed the image into the latent space, we introduce optimization-based GAN inversion, which adopts identity loss to improve the identity similarity between the input image and the embedded image. Furthermore, we add delta-regularization to the loss function to improve editability, which explicitly enforces the latent vector proximity to the original latent space. We evaluate our GAN inversion method on CelebA-HQ dataset. The embedded image quality of our method outperforms other state-of-the-art methods, especially our improvement attains 30% in terms of identity similarity. We show our contributions to various loss design choices by ablation study. We also show that adding delta-regularization increases the editing capability of the embedded images qualitatively. For face reenactment, we build the mapping network, which learns to map the displacement in the facial landmark images of the source and driving images to the displacement in the latent vectors of the source and driving images. We evaluate our face reenactment method on MEAD dataset. Our approach generates much higher-resolution videos compared to other state-of-the-art face reenactment methods by adopting the pretrained image generator. Our method generates the reenacted images with well-preserved source identity and synced with the driving facial movements. Our method achieves the highest SSIM and the lowest LMD for self-reenactment and outperforms existing approaches in ID similarity, ARD, and AUH for cross-identity reenactment. Our model achieved this without requiring any 3D modeling or a number of source images. Instead, it is trained on a large collection of videos and learns to map the facial landmarks to the latent vector of StyleGAN which is a high-resolution image generator.

As for the negative social impacts of our method, it can be used to create deepfake if not timely controlled. It is a kind of serious exploitation of machine learning techniques used to create the illusion that someone said something that they didn't say, and such changes are not noticeable to the human eye, compelling people to believe easily. However, in controlled settings, our method can be used for many positive creative purposes. Specifically, it can be used for driving characters in computer games, dubbing in movies, animating avatars for virtual assistants, and telecommunications. With each

passing day, the world's growing digital landscape presents numerous possibilities, and face reenactment is among the innovations that offer many opportunities. It is important for users of these kinds of works to utilize them in an ethical manner. We believe that these works will bring benefits and minimize the need for manual effort in the realm of professional content creation.

Abbreviations

GAN Generative Adversarial Network

StyleGAN Style-Based Generative Adversarial Network

DCGAN Deep Convolutional Generative Adversarial Network

CNN Convolutional Neural Network

CelebA-HQ High-Quality version of CelebFaces Attributes

PGGAN Progressive Growing Generatie Adversarial Network

FFHQ Flickr-Faces HQ

AdaIN Adaptive Instance Normalization

FID Fréchet Iception Distance

PPL Perceptual Path Length

I2S Image2StyleGAN

I2S++ Image2StyleGAN++

pSp Pixel2Style2Pixel

FPN Feature Pyramid Network

e4e Encoder for Editing

FSE Featyre-Style Encoder

PC-AVS Pose-Controllable Audio- Visual System

SPACE Speech-driven Portrait Animation with Controllable Expression

FOMM First Order Motion Model

OSFV One-Shot Free-View neural talking head synthesis

MRAA Motion Representations for Articuated Animation

DaGAN Depth-aware GAN

PECHead Pose and Expression Controllable Head model

MSE Mean Squared Error

SSIM Structural Similarity Index Measure

MSSIM Mean Structural Similarity Index Measure

FSIM Feature Similarity

LPIPS Learned Perceptual Image Patch Similarity

ArcFace Additive Angular Margin Loss

PSNR Peak Signal-to-Noise Raito

MEAD Multi-view Emotional Audio-visual Dataset

LMD Landmark Distance

ARD Average Rotation Distance

AUH Action Unit Hamming distance

AKD Average Keypoint Distance

AFHQ Animal Faces-HQ

LSTM Long Short-Term Memory

List of Figures

List of Tables

Bibliography

[Aga+22] M. Agarwal, R. Mukhopadhyay, V. Namboodiri, and C. V. Jawahar. *Audio-Visual Face Reenactment*. 2022. arXiv: 2210.02755 [cs.CV].

[AQW19a] R. Abdal, Y. Qin, and P. Wonka. "Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?" In: *CoRR* abs/1904.03189 (2019). arXiv: 1904.03189.

[AQW19b] R. Abdal, Y. Qin, and P. Wonka. "Image2StyleGAN++: How to Edit the Embedded Images?" In: *CoRR* abs/1911.11544 (2019). arXiv: 1911.11544.

[AQW19c] R. Abdal, Y. Qin, and P. Wonka. *Official implementation of image2stylegan: how to embed images into the stylegan latent space?* https://github.com/zaidbhat1234/Image2StyleGAN. 2019.

[BDS18] A. Brock, J. Donahue, and K. Simonyan. "Large Scale GAN Training for High Fidelity Natural Image Synthesis." In: *CoRR* abs/1809.11096 (2018). arXiv: 1809.11096.

[BT17] A. Bulat and G. Tzimiropoulos. "How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230, 000 3D facial landmarks)." In: *CoRR* abs/1703.07332 (2017). arXiv: 1703.07332.

[Bur+20] E. Burkov, I. Pasechnik, A. Grigorev, and V. S. Lempitsky. "Neural Head Reenactment with Latent Pose Descriptors." In: *CoRR* abs/2004.12000 (2020). arXiv: 2004.12000.

[Cao+19] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[Cha+19] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros. "Everybody Dance Now." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[Cho+19] Y. Choi, Y. Uh, J. Yoo, and J. Ha. "StarGAN v2: Diverse Image Synthesis for Multiple Domains." In: *CoRR* abs/1912.01865 (2019). arXiv: 1912.01865.

[CJZ17] J. S. Chung, A. Jamaludin, and A. Zisserman. "You said that?" In: *CoRR* abs/1705.02966 (2017). arXiv: 1705.02966.

[CZ16] J. S. Chung and A. Zisserman. "Out of time: automated lip sync in the wild." In: *Workshop on Multi-view Lip-reading, ACCV*. 2016.

[DGZ18] J. Deng, J. Guo, and S. Zafeiriou. "ArcFace: Additive Angular Margin Loss for Deep Face Recognition." In: *CoRR* abs/1801.07698 (2018). arXiv: 1801.07698.

[DZS21] M. C. Doukas, S. Zafeiriou, and V. Sharmanska. *HeadGAN: One-shot Neural Head Synthesis and Editing*. 2021. arXiv: 2012.08261 [cs.CV].

[Gao+23] Y. Gao, Y. Zhou, J. Wang, X. Li, X. Ming, and Y. Lu. *High-Fidelity and Freely Controllable Talking Head Video Generation*. 2023. arXiv: 2304.10168 [cs.CV].

[Goo+14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML].

[Gur+22] S. Gururani, A. Mallya, T.-C. Wang, R. Valle, and M.-Y. Liu. *SPACE: Speech-driven Portrait Animation with Controllable Expression*. 2022. arXiv: 2211.09809 [cs.CV].

[Heu+17] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, G. Klambauer, and S. Hochreiter. "GANs Trained by a Two Time-Scale Update Rule Converge to a Nash Equilibrium." In: *CoRR* abs/1706.08500 (2017). arXiv: 1706.08500.

[Hon+22a] F.-T. Hong, L. Zhang, L. Shen, and D. Xu. *Depth-Aware Generative Adversarial Network for Talking Head Video Generation*. 2022. arXiv: 2203.06605 [cs.CV].

[Hon+22b] F.-T. Hong, L. Zhang, L. Shen, and D. Xu. *Official implementation of depth-aware generative adversarial network for talking head video generation*. https://github.com/harlanhong/CVPR2022-DaGAN. 2022.

[JAF16] J. Johnson, A. Alahi, and L. Fei-Fei. "Perceptual Losses for Real-Time Style Transfer and Super-Resolution." In: *CoRR* abs/1603.08155 (2016). arXiv: 1603.08155.

[Ji+22] X. Ji, H. Zhou, K. Wang, Q. Wu, W. Wu, F. Xu, and X. Cao. *EAMM: One-Shot Emotional Talking Face via Audio-Based Emotion-Aware Motion Model*. 2022. arXiv: 2205.15278 [cs.CV].

[Kar+17] T. Karras, T. Aila, S. Laine, and J. Lehtinen. "Progressive Growing of GANs for Improved Quality, Stability, and Variation." In: *CoRR* abs/1710.10196 (2017). arXiv: 1710.10196.

[Kar+19] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. "Analyzing and Improving the Image Quality of StyleGAN." In: *CoRR* abs/1912.04958 (2019). arXiv: 1912.04958.

[Kar+20] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila. "Training Generative Adversarial Networks with Limited Data." In: *CoRR* abs/2006.06676 (2020). arXiv: 2006.06676.

[Kar+21a] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila. "Alias-Free Generative Adversarial Networks." In: *CoRR* abs/2106.12423 (2021). arXiv: 2106.12423.

[Kar+21b] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila. *Official implementation of alias-free generative adversarial networks*. https://github.com/NVlabs/stylegan3. 2021.

[KB17] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[KLA18] T. Karras, S. Laine, and T. Aila. "A Style-Based Generator Architecture for Generative Adversarial Networks." In: *CoRR* abs/1812.04948 (2018). arXiv: 1812.04948.

[KS14] V. Kazemi and J. Sullivan. "One millisecond face alignment with an ensemble of regression trees." In: *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 1867–1874.

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C. Burges, L. Bottou, and K. Weinberger. Vol. 25. Curran Associates, Inc., 2012.

[Li+17] Y. Li, N. Wang, J. Liu, and X. Hou. "Demystifying Neural Style Transfer." In: *CoRR* abs/1701.01036 (2017). arXiv: 1701.01036.

[Lin+16] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. "Feature Pyramid Networks for Object Detection." In: *CoRR* abs/1612.03144 (2016). arXiv: 1612.03144.

[Nar+20] J. Naruniec, L. Helminger, C. Schroers, and R. M. Weber. "High-Resolution Neural Face Swapping for Visual Effects." In: *Computer Graphics Forum* (2020). ISSN: 1467-8659. DOI: 10.1111/cgf.14062.

[Pas+19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." In: *CoRR* abs/1912.01703 (2019). arXiv: 1912.01703.

[Pat+21a] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski. *Official implementation of StyleCLIP: text-driven manipulation of StyleGAN imagery*. https://github.com/orpatashnik/StyleCLIP. 2021.

[Pat+21b] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski. *StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery*. 2021. arXiv: 2103.17249 [cs.CV].

[Pra+20a] K. R. Prajwal, R. Mukhopadhyay, V. P. Namboodiri, and C. V. Jawahar. "A Lip Sync Expert Is All You Need for Speech to Lip Generation In The Wild." In: *CoRR* abs/2008.10010 (2020). arXiv: 2008.10010.

[Pra+20b] K. R. Prajwal, R. Mukhopadhyay, J. Philip, A. Jha, V. P. Namboodiri, and C. V. Jawahar. "Towards Automatic Face-to-Face Translation." In: *CoRR* abs/2003.00418 (2020). arXiv: 2003.00418.

[Ric+20a] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or. "Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation." In: *CoRR* abs/2008.00951 (2020). arXiv: 2008.00951.

[Ric+20b] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or. *Official implementation of encoding in style: a stylegan encoder for image-to-image translation*. https://github.com/eladrich/pixel2style2pixel. 2020.

[RMC16] A. Radford, L. Metz, and S. Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2016. arXiv: 1511.06434 [cs.LG].

[Saj+18] M. S. M. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly. *Assessing Generative Models via Precision and Recall*. 2018. arXiv: 1806.00035 [stat.ML].

[She+20] Y. Shen, C. Yang, X. Tang, and B. Zhou. "InterFaceGAN: Interpreting the Disentangled Face Representation Learned by GANs." In: *TPAMI* (2020).

[Sia+18] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe. "Animating Arbitrary Objects via Deep Motion Transfer." In: *CoRR* abs/1812.08861 (2018). arXiv: 1812.08861.

[Sia+19a] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe. "First Order Motion Model for Image Animation." In: *Conference on Neural Information Processing Systems (NeurIPS)*. Dec. 2019.

[Sia+19b] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe. *Official implementation of first order motion model for image animation*. `https://github.com/AliaksandrSiarohin/first-order-model`. 2019.

[Sia+21a] A. Siarohin, O. J. Woodford, J. Ren, M. Chai, and S. Tulyakov. "Motion Representations for Articulated Animation." In: *CoRR* abs/2104.11280 (2021). arXiv: `2104.11280`.

[Sia+21b] A. Siarohin, O. J. Woodford, J. Ren, M. Chai, and S. Tulyakov. *Official implementation of motion representations for articulated animation*. `https://github.com/snap-research/articulated-animation`. 2021.

[Sim+17] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. "Hand Keypoint Detection in Single Images using Multiview Bootstrapping." In: *CVPR*. 2017.

[SZ21] Y. Shen and B. Zhou. "Closed-Form Factorization of Latent Semantics in GANs." In: *CVPR*. 2021.

[Sze+15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. *Rethinking the Inception Architecture for Computer Vision*. 2015. arXiv: `1512.00567 [cs.CV]`.

[Tov+21a] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or. "Designing an Encoder for StyleGAN Image Manipulation." In: *CoRR* abs/2102.02766 (2021). arXiv: `2102.02766`.

[Tov+21b] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or. *Official implementation of designing an encoder for stylegan image manipulation*. `https://github.com/omertov/encoder4editing`. 2021.

[Wan+04] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. "Image Quality Assessment: From Error Visibility to Structural Similarity." In: *Image Processing, IEEE Transactions on* 13 (May 2004), pp. 600–612. DOI: `10.1109/TIP.2003.819861`.

[Wan+19] T.-C. Wang, M.-Y. Liu, A. Tao, G. Liu, J. Kautz, and B. Catanzaro. *Few-shot Video-to-Video Synthesis*. 2019. arXiv: `1910.12713 [cs.CV]`.

[Wan+20] K. Wang, Q. Wu, L. Song, Z. Yang, W. Wu, C. Qian, R. He, Y. Qiao, and C. C. Loy. "MEAD: A Large-scale Audio-visual Dataset for Emotional Talking-face Generation." In: *ECCV*. 2020.

[Wan+21] S. Wang, L. Li, Y. Ding, C. Fan, and X. Yu. *Audio2Head: Audio-driven One-shot Talking-head Generation with Natural Head Motion*. 2021. arXiv: `2107.09293 [cs.CV]`.

[WKZ18a] O. Wiles, A. S. Koepke, and A. Zisserman. *Official implementation of X2Face: a network for controlling face generation by using images, audio, and pose codes*. `https://github.com/oawiles/X2Face`. 2018.

[WKZ18b] O. Wiles, A. S. Koepke, and A. Zisserman. *X2Face: A network for controlling face generation by using images, audio, and pose codes*. 2018. arXiv: `1807.10550 [cs.CV]`.

[WML21] T.-C. Wang, A. Mallya, and M.-Y. Liu. "One-Shot Free-View Neural Talking-Head Synthesis for Video Conferencing." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2021.

[WSB03] Z. Wang, E. Simoncelli, and A. Bovik. "Multiscale structural similarity for image quality assessment." In: *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*. Vol. 2. 2003, 1398–1402 Vol.2. DOI: `10.1109/ACSSC.2003.1292216`.

[Xia+21] W. Xia, Y. Zhang, Y. Yang, J. Xue, B. Zhou, and M. Yang. "GAN Inversion: A Survey." In: *CoRR* abs/2101.05278 (2021). arXiv: `2101.05278`.

[Yao+21] G. Yao, Y. Yuan, T. Shao, S. Li, S. Liu, Y. Liu, M. Wang, and K. Zhou. "One-shot Face Reenactment Using Appearance Adaptive Normalization." In: *CoRR* abs/2102.03984 (2021). arXiv: `2102.03984`.

[Yao+22a] X. Yao, A. Newson, Y. Gousseau, and P. Hellier. "Feature-Style Encoder for Style-Based GAN Inversion." In: *CoRR* abs/2202.02183 (2022). arXiv: `2202.02183`.

[Yao+22b] X. Yao, A. Newson, Y. Gousseau, and P. Hellier. *Official implementation of feature-style encoder for style-based gan inversion*. `https://github.com/InterDigitalInc/FeatureStyleEncoder`. 2022.

[Zak+20] E. Zakharov, A. Ivakhnenko, A. Shysheya, and V. Lempitsky. *Fast Bi-layer Neural Synthesis of One-Shot Realistic Head Avatars*. 2020. arXiv: `2008.10174 [cs.CV]`.

[Zha+11] L. Zhang, L. Zhang, X. Mou, and D. Zhang. "FSIM: A Feature Similarity Index for Image Quality Assessment." In: *IEEE Transactions on Image Processing* 20.8 (2011), pp. 2378–2386. DOI: `10.1109/TIP.2011.2109730`.

[Zha+18] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric." In: *CoRR* abs/1801.03924 (2018). arXiv: `1801.03924`.

[Zha+19] Y. Zhang, S. Zhang, Y. He, C. Li, C. C. Loy, and Z. Liu. "One-shot Face Reenactment." In: *CoRR* abs/1908.03251 (2019). arXiv: `1908.03251`.

[Zho+20] Y. Zhou, D. Li, X. Han, E. Kalogerakis, E. Shechtman, and J. Echevarria. "MakeItTalk: Speaker-Aware Talking Head Animation." In: *CoRR* abs/2004.12992 (2020). arXiv: 2004.12992.

[Zho+21] H. Zhou, Y. Sun, W. Wu, C. C. Loy, X. Wang, and Z. Liu. "Pose-Controllable Talking Face Generation by Implicitly Modularized Audio-Visual Representation." In: *CoRR* abs/2104.11116 (2021). arXiv: 2104.11116.

[ZZ22] J. Zhao and H. Zhang. *Thin-Plate Spline Motion Model for Image Animation*. 2022. arXiv: 2203.14367 [cs.CV].