

# Face Reenactment with Generative Landmark Guidance



Prifysgol Abertawe  
Swansea University

**Chen Hu**

School of Mathematics and Computer Science  
Swansea University

This dissertation is submitted for the degree of  
*MSc by Research on Visual Computing*

December 2022

## Abstract

Face reenactment is a task aiming for transferring the expression and head pose from one face image to another. Recent studies mainly focus on estimating optical flows to warp input images' feature maps to reenact expressions and head poses in synthesized images. However, the identity preserving problem is one of the major obstacles in these methods. The problem occurs when the model fails to preserve the detailed information of the source identity, namely the identity of the face we wish to synthesize, and especially obvious when reenacting different identities. The underlying factors may include unseen the leaking of driving identity. The driving identity stands for the identity of the face that provides the desired expression and head pose. When the source and the driving hold different identities, the model tends to mix the driving's facial features with those of the source, resulting in inaccurate optical flow estimation and subsequently causing the identity of the synthesized face to deviate from the source.

In this paper, we propose a novel face reenactment approach via generative landmark coordinates. Specifically, a conditional generative adversarial network is developed to estimate reenacted landmark coordinates for the driving image, which successfully excludes its identity information. We then use generated coordinates to guide the alignment of individually reenacted facial landmarks. These coordinates are also injected into the style transferal module to increase the realism of face images. We evaluated our method on the VoxCeleb1 dataset for self-reenactment and the CelebV dataset for reenacting different identities. Extensive experiments demonstrate that our method can produce realistic reenacted face images by lowering the error in head pose and enhancing our models' identity preserving capability.

In addition to the conventional centralized learning, we deployed our model and used the CelebV dataset for federated learning in an aim to mitigate potential privacy issues involved in research on face images. We show that the proposed method is capable of showing competitive performance in the setting of federated learning.

# Declarations

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed  Date 23/1/2023

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed  Date 23/1/2023

I hereby give consent for my thesis, if accepted, to be available for electronic sharing

Signed  Date 23/1/2023

The University's ethical procedures have been followed and, where appropriate, that ethical approval has been granted.

Signed  Date 23/1/2023



# Contents

<b>List of Figures</b>	<b>VI</b>
<b>List of Tables</b>	<b>VII</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Contributions . . . . .	3
1.3 Thesis Layout . . . . .	4
1.4 Publication . . . . .	5
<b>2 Background</b>	<b>6</b>
2.1 Face Reenactment . . . . .	6
2.1.1 Rendering-based Face Reenactment . . . . .	6
2.1.2 Image Warping-based Face Reenactment . . . . .	7
2.2 Generative Adversarial Networks . . . . .	11
2.2.1 Vanilla GAN and Conditional GAN . . . . .	11
2.2.2 Image-to-Image Translation and Video Generation . . . . .	12
2.3 Vision Transformer . . . . .	13
2.4 Federated Learning . . . . .	14
2.5 Summary . . . . .	15
<b>3 Face Reenactment with Generative Landmark Guidance</b>	<b>17</b>
3.1 Optical Flow Estimation . . . . .	18
3.2 Individual Landmark Reenactment . . . . .	19
3.3 Landmark Estimation . . . . .	21
3.3.1 Landmark Style Transfer . . . . .	21
3.3.2 Landmark Conditional GAN . . . . .	21
3.4 Image Generator . . . . .	23
3.5 Loss Function . . . . .	24
3.6 Application: Federated Face Reenactment . . . . .	25
3.6.1 Federated Learning Configurations . . . . .	26
3.6.2 Model Aggregation . . . . .	27
3.7 Summary . . . . .	28

<b>4</b>	<b>Experiments</b>	<b>32</b>
4.1	Datasets and Experimental Settings . . . . .	32
4.2	Metrics . . . . .	33
4.3	Experimental Results and Analysis . . . . .	34
4.3.1	Self-reenactment . . . . .	35
4.3.2	Reenacting Different Identities . . . . .	36
4.3.3	ViT vs ResNet . . . . .	37
4.3.4	Landmark Estimation and Style Transfer . . . . .	38
4.3.5	Federated Learning vs Centralized Learning . . . . .	40
<b>5</b>	<b>Conclusion</b>	<b>41</b>
5.1	Limitations . . . . .	42
5.2	Future Work . . . . .	43
	<b>References</b>	<b>45</b>

# List of Figures

1.1	Examples of Face Reenactment. . . . .	1
3.1	Overview of proposed method for self-reenactment and reenacting different identities. Dashed boxes show loss functions that are responsible for the corresponding module. . . . .	17
3.2	Architecture of Facial Feature Extraction Module. (a) Extracting features from input images; (b) Estimating the optical flow based on extracted features. . . . .	18
3.3	Individual landmark reenactment and examples of landmark generation. (a) Individual landmarks are concurrently reenacted with models that share the same architecture. (b) Example of mouth reenactment showing the architecture of the landmark reenactment model. (c) Landmark coordinates estimated by style transfer significantly sacrifice head pose accuracy for identity preserving, whereas landmark conditional GAN better balances this trade off. . . . .	20
3.4	Facial action units (AUs) for estimating landmarks and evaluating performance. . . . .	22
3.5	The architecture of landmark coordinate conditional GAN. . . . .	22
3.6	The style transfer branch and the image generator. . . . .	23
3.7	Configuration of Federated Face Reenactment. . . . .	26
3.8	Architecture of Client Models. . . . .	26
3.9	Qualitative results of proposed models on CelebV dataset. . . . .	30
3.10	Qualitative results of proposed models on CelebV dataset. . . . .	31
4.1	Self-reenactment on VoxCeleb1 dataset. . . . .	35
4.2	Comparison of Reenacting Different Identities on CelebV. . . . .	38
4.3	Optical flow combined with style transfer improved the quality of generated images. . . . .	39
4.4	Comparison between federated model and centralized model. . . . .	40

## List of Tables

2.1	Comparison of Landmark Coordinate Reenactment Methods . . . . .	11
4.1	Details of datasets for evaluated tasks. . . . .	32
4.2	Evaluation of Self-reenactment on VoxCeleb1 . . . . .	36
4.3	Evaluation of Reenacting Different Identities with Unseen Data on CelebV .	36
4.4	Evaluation of Reenacting Different Identities with Models Trained on CelebV	37
4.5	Evaluation of Landmark Estimation for Reenacting Different Identities on CelebV . . . . .	38
4.6	Evaluation of Style Transfer for Reenacting Different Identities on CelebV .	39
4.7	Evaluation of Federated Face Reenactment on CelebV . . . . .	40



# 1 Introduction

## 1.1 Motivation

Deep learning has achieved great success in a variety of computer vision tasks, ranging from image classification to video understanding. Face reenactment, a conditional image generation task, also benefits from deep learning. The input, namely the condition, to a face reenactment model comprises two parts, the source and the driving. The source is one or a set of images of a specific person, serving to provide appearance features of the person. The driving image could be the face of the same person or other people. The goal of face reenactment is to transfer the head pose and expression from the driving image to the face in the source image, as shown in the examples of Figure 1.1. Real world application of face reenactment includes video conferencing and film production. In the scenario video conferencing, the speaker’s face can be reenacted to match the face motion of a translator [1]. In the film industry, face reenactment can be used in a similar fashion, creating more natural localized motion pictures in different languages. Film makers can also further edit actors expressions without re-shooting the entire scene.



Figure 1.1: Examples of Face Reenactment.

Given the fact that it is infeasible to collect image pairs of two different people with identical head pose and expression, the self-supervised training strategy proposed by authors of X2Face [2] greatly helped the evolution of face reenactment methods. The self-supervised training of face reenactment constrains the identity of an input source-driving image pair to be the same person during training, therefore the driving image is also the groundtruth for the generated image. Despite the ease of training, in the testing scenario where the source and driving image are taken from different people, models trained by this strategy is

at the risk of mixing the driving’s identity in the image generator, resulting in the identity preserving problem, that is, the reenacted image shares structural similarity with both people in the input, instead of being the exact person in the source image.

Location of facial landmarks is valuable for defining a person’s identity and head pose. During self-supervised training, if the eyes, nose and mouth in an generated image are precisely aligned with their corresponding location in the driving image, the generated image is more likely to be a faithful reenactment. Landmark locations can then be used to guide the model in the self-reenactment scenario. However, when reenacting different people, landmark locations in the driving image does not lead to desired output, as the location now reflects facial features of a different person, which can only aggravate the identity preserving problem. To help face reenactment methods benefit from landmark locations, landmark coordinates also need to be reenacted. If these coordinates reflect the source’s identity while matching the driving’s head pose and expression, they can guide the model to process the test sample as if it is a self-reenactment case.

Motivated by the above observation, we propose a face reenactment method that explicitly leverages landmark locations to guide the reenactment process. To estimate landmark coordinates that meet the requirements of face reenactment, namely reenacting the driving’s expression while preserving the source’s identity, we first adopt a coordinate style transfer method to estimate reenacted landmark coordinates. To further enhance the quality of generated images, we then introduce a generative network that takes the source image’s identity, the driving image’s head pose angle, and facial action units detected in the driving image as input, more accurately estimating reenacted landmark coordinates. We also reenact crucial facial landmarks individually, and align them based on estimated coordinates. When reenacting different identities, we inference style transfer parameters based on estimated landmark coordinates to correct distortion in generated images.

Since face reenactment is a task that involves the entire region of faces, it requires models to extract features that cover the structure and appearance of the entire face. Vision Transformer is a particular neural network architecture which is based on the self-attention mechanism and designed to perceive the interconnections of all regions in an image. We argue that Vision Transformer is also a suitable backbone for face reenactment, however, to the best of our knowledge, little research has been conducted on face reenactment with Vision Transformer. Therefore, another focus of the thesis is to investigate the possibility of applying Vision Transformer to face reenactment.

The nature of this study requires the use of human face images, however, there has

been a growing concern on privacy issues involved in computer vision research. Face images are unique biometric data that malicious attackers are constantly trying to steal and leverage for illegal purposes. Therefore, faces in the ImageNet dataset [3] has been obfuscated while the MS-Celeb-1M dataset [4] has been retracted from the internet [5]. To bridge the gap between research and privacy protection, federated learning was proposed. This deep learning paradigm allows researchers to leave the private data alone with data owners, only distributedly trained models are transferred to researchers for evaluation. In addition to the proposed method, we demonstrate that our model is compatible with federated learning, thus the privacy issues can be mitigated.

## 1.2 Contributions

Contributions of the thesis are the following:

- We propose a face reenactment method guided by facial landmark coordinates. An optical flow is first estimated based on the input source-driving image pair, the estimated optical flow is subsequently used to warp the feature maps of source images, then individual landmarks are respectively reenacted and aligned based on landmark coordinates to guide the warped results. Direct guidance from landmark coordinates ensures lower head pose error compared to existing methods.
- We introduce a landmark conditional GAN to alleviate the identity preserving problem induced by the self-supervised learning strategy in recent face reenactment methods. The proposed landmark GAN generates landmark coordinates based on the input source person’s identity, desired head pose and facial action units, generated landmark coordinates are subsequently used to guide the face reenactment process. Because facial action units are a widely used building blocks for human expressions, our landmark conditional GAN may be beneficial for face image generation controlled by expressions. In addition, we estimate style transfer parameters based on the generated landmark coordinates to improve the realism of generated faces. Since the driving’s identity has been explicitly excluded from the proposed landmark GAN, we greatly improved our models’ performance on identity preserving.
- We apply the FedGAN algorithm and the CelebV dataset for training face reenactment models in the federated setting. Considering the rise of privacy concerns in deep learning research, federated learning is becoming more valuable for projects involving

sensitive biometric data. Our work will provide the stepping stone for further research on federated face reenactment.

- We evaluate our method on the VoxCeleb1 [6] dataset for self-reenactment and the CelebV [7] dataset for reenacting different identities. Experiments on the CelebV dataset covers three face reenactment scenarios: reenacting different unseen identities, reenacting different known identities and reenacting known identities with models trained through federated learning.

### 1.3 Thesis Layout

This rest of the thesis is split up into the following chapters:

- **Chapter 2** starts with a brief introduction on neural network architectures involved in our method, including convolutional neural networks, transformers, and generative adversarial networks. We then introduce the development of recent face reenactment methods, from 3D rendering to self-supervised 2D methods, which are more commonly used nowadays. Lastly, we present an introduction on federated learning, specifically, we focus on the FedAvg algorithm as it laid the foundation for our experiments.
- **Chapter 3** presents each module in our proposed neural network. We first introduce how we leverage the Vision Transformer to estimate optical flows for warping the feature maps of source images, we then explain the reenactment of individual facial landmarks and the method we used to estimate landmark coordinates for accurately aligning reenacted landmarks. In this chapter we show the architecture of our image generator and how the style transfer branch is blended in to help generate more realistic images. Definitions of loss functions used in this study are also introduced in this chapter. In addition to the architectural details of the proposed method, we show how this model is deployed and trained in the federated learning setting.
- **Chapter 4** focuses on presenting experimental results and analysis on these results. We give definitions of metrics for evaluating face reenactment methods, and how datasets are split up for different scenarios. For centralized learning, we evaluate our models through 3 different experimental settings: self-reenactment, reenacting different unseen identities, and reenacting different known identities. In terms of federated learning, we focus on reenacting different known identities to evaluate how our models perform compared to those trained through centralized learning.

- **Chapter 5** concludes the thesis and gives analysis on limitations of the proposed method, we also discuss potential solutions and areas that may bring improvement to our method.

## 1.4 Publication

The following is a list of published and submitted papers as a result of this thesis:

Chen Hu, Xianghua Xie. One-Shot Decoupled Face Reenactment with Vision Transformer. Pattern Recognition and Artificial Intelligence (ICPRAI) 2022. Lecture Notes in Computer Science, vol 13364. Springer, Cham.

Chen Hu, Xianghua Xie, Ling Wu. Face Reenactment with Generative Landmark Guidance. Image and Vision Computing, 2022.

## 2 Background

In this chapter, we begin by outlining two types of face reenactment methods: rendering-based and more recent image warping-based. We focus on introducing image warping-based methods as they are more widely used nowadays due to the ease of training and better generalization capability, our method also falls into this category. We then give an overview on generative adversarial networks, which are an essential component of ours and many other face reenactment methods. We also briefly review the Vision Transformer architecture, which is used in our method in hope of helping our model better perceive human faces. Lastly, we cover federated learning, explaining its definition and noteworthy algorithms which are the cornerstone of our research in this area.

### 2.1 Face Reenactment

Face reenactment in general is not limited to visual input, audio-driven face reenactment [8,9] is also an active research area. These methods often learn a mapping from audio features to facial blendshapes, then reenacted images can be rendered based on tweaked blendshapes that match input audio track. The most noticeable difference between audio-driven and visual-driven face reenactment is that audio-driven face reenactment focuses on reenacting the mouth region of the face image, while visual-driven face reenactment needs to consider the expression over the entire face and the movement of the head. Audio-driven methods are beyond the scope of the thesis, we instead focus on presenting an introduction on visual-driven face reenactment methods.

Approaches to visual-driven face reenactment can be categorized by the way of synthesizing new images, namely rendering from 3D models or warping 2D images. In this chapter we look at typical examples of each type of face reenactment method and discuss how they are related to our method.

#### 2.1.1 Rendering-based Face Reenactment

Early face reenactment studies [1, 10–13] prefer rendering desired images from estimated 3D face models. The pipeline of rendering-based methods generally involves fitting faces from images to 3D models, then morphing these 3D faces and rendering the reenacted results. For instance, given a pair of source and driving image, Deep Video Portraits [11] first estimates 3DMM [14] faces for both the source and driving images. Estimated 3D models contain the illumination, identity, pose, expression and eye gaze parameters. By plugging in the

source images’ illumination and identity parameters into the driving images’ 3D models, the reenacted 3D faces are obtained, which can then be rendered into desired images. Deep Video Portraits is capable of generating convincing images, but it has high computation cost and difficulties with changing poses. The rendering process of this method has two stages, a 3DMM face with 53,215 vertices is first rendered based on estimated illumination and reflectance parameters, the rendered output is then sent to an image translation network to synthesize the reenacted image. Regarding the problem with pose changes, since this method only focuses on the heads, undesired artifacts can be seen around the upper body when there are obvious pose changes in input images.

Face2Face [1] is another example of rendering-based face reenactment methods. The authors first reconstruct 3D models for the source and driving person, respectively. Expression features are then extracted from the driving model and transferred to the source model. The reenacted 3D face is obtained in a similar fashion like Deep Video Portraits, namely plugging the source parameters into the driving 3D model. To further enhance the generated image, the authors retrieve the RGB mouth region from the entire video sequence of the source person, aiming to find the frame in which the mouth movement best matches that of the driving frame. Lastly, the final output is rendered based on the retrieved mouth and reenacted 3D model. Retrieving the mouth region from the input sequence is the most noticeable issue with Face2Face, as it requires the input to be diverse and long enough to provide good reenactment around the mouth.

In general, rendering-based methods can synthesize high-fidelity images. However, to capture facial details in the 3D space, the number of vertices has to increase, which creates more computation overhead. In addition, to further enhance rendered images, rendering based methods such as Face2Face resort to retrieving information from input video sequences, resulting in the disadvantage of not being able to process short videos or static images. The above limitations may impede real-world applications of rendering-based methods [15], recent research focuses more on image warping-based methods instead.

### 2.1.2 Image Warping-based Face Reenactment

Recent works [2, 15–20] propose one-shot or few-shot face reenactment and utilise optical flow to map pixels from the source image to the reenacted image, image warping then becomes an essential operation for these methods. Image warping on convolutional neural networks (CNN) was first proposed in [21], where the model can estimate an optical flow that warps skewed numerical digit back to the regular view, thus improving the classification accuracy.

The estimated optical flow is usually a tensor with shape  $H \times W \times 2$ , where  $H$  is the height of the image,  $W$  is the width of the image. One channel of the optical flow tensor contains the X-axis coordinates for the warping operation, the other channel stores the values on the Y-axis. For face reenactment, the optical flow is first estimated from features extracted from both the source and driving image, then image warping is conducted, which uses the optical flow to determine how pixel values should be sampled from the source image or its feature maps such that the desired image can be generated.

In X2Face, the authors first estimate an optical flow that warps the source image to obtain the embedded face, namely warping the input source image of arbitrary head pose to its corresponding frontal view. Another optical flow is estimated from the driving image, and this optical flow is applied to the embedded face, leading to the reenacted output of X2Face model. Directly applying optical flow on the face image is relatively stable when the pose variation between the source and driving image is subtle. When there are drastic changes in the head pose, warping the RGB image can often distort the reenacted face, resulting in degraded output. More recent methods [15–18] choose to estimate the optical flow for intermediate feature maps of input images. The benefit of this choice is that even distortion happens in intermediate feature maps, subsequent convolution layers can still learn to counter this effect, whereas in the case of X2Face, once the RGB image is warped, nothing can further enhance the output image.

As mentioned in Chapter 1, obtaining images for different people with the exact same poses and expressions is infeasible in practice, a now widely adopted self-supervised learning paradigm was proposed in [2]. Given the source image sampled from a video sequence, a corresponding driving image of the same person is randomly chosen from the same video, making supervised learning possible as the driving image is exactly the expected reenactment result.

Although the self-supervised strategy in X2Face enables easy training for face reenactment models, it introduces a more noticeable problem, namely the identity preserving problem first described in [16]. When the identity of the source image and that of the driving are different from each other, models trained by self-supervised learning tend to combine the identity features of both identities, the generated face will fail to preserve the identity of the source image. To remedy this issue, the work of [15, 16, 19] resorts to facial landmark coordinates as a cue to help the model retain the source’s identity. In contrast, MonkeyNet [18] and the FirstOrder [17] method focus on animating arbitrary images in stead of face reenactment, these methods does not explicitly leverage facial landmark points, they



detect key points in input images and then derive the optical flow based on the motion of key points. When facial landmark coordinates are used to guide the reenactment process, these coordinates also need be reenacted to match the identity of the source face along with the head pose and the expression of the driving face.

The authors of MeshGCN [15] explicitly estimate the dense face coordinates with the help of 3D Morphable Models (3DMM) [14]. 3DMM decouples a 3D face into the following three components: an average emotionless 3D face, identity displacement and expression displacement. The 3DMM formulation enables straightforward face reenactment in the 3D space. This is because the source face’s 3DMM identity parameters can be extracted from the input source image, while the driving face’s expression parameters can be also extracted in a similar fashion. Then by replacing the expression parameters of the source 3DMM face with corresponding parameters extracted from the driving image, the 3D face with the source identity and the driving expression can be obtained.

The authors of MeshGCN first estimate 3DMM parameters from input images, then they construct reenacted 3D faces following the above process. To estimate the optical flow for the warping operation, the authors apply a graph neural network [22] for this task. The input to this network is the dense coordinates of the source model  $V_{reen} \in \mathbb{R}^{53215 \times 3}$  and that of the reenacted model  $V_{reen} \in \mathbb{R}^{53215 \times 3}$ , incurring high computation cost. However, due to the decoupling of identity and expression in 3DMM’s formulation, the driving face’s identity is removed from the reenactment process, which greatly helps MeshGCN to achieve excellent performance on identity preserving.

The authors of MarioNETte [16] are also inspired by 3DMM, but their method estimates 3D coordinates of 68 landmark points instead of dense coordinates like MeshGCN. The authors decomposes landmark points following the 3DMM formulation, they first perform principal component analysis (PCA) on landmark coordinates extracted from the VoxCeleb1 [6] dataset, the principal components are used as identity and expression basis. The authors then train a neural network to regress coefficient associated with the expression basis such that their landmark decomposition matches the landmark coordinates in the input image. The landmark reenacted of MarioNETte is also achieved by replacing the source face’s expression parameters with that of the driving face’s. Once the reenacted landmark coordinates are computed, the authors of MarioNETte use the coordinates to generate rasterized landmark images, the optical flows are estimated based on these landmark images combined with the original input. Nonetheless, the performance of MarioNETte is limited by the expressiveness of principal components, and MarioNETte requires multiple images of

the source identity as input for optimal performance.

In addition to above methods, there are also face reenactment methods that do not reenact landmark coordinates. NeuralHead [23] is an example of such method. The authors directly feed the facial landmark heatmap of the driving image into the image generator, source images are sent to a neural network to predict style transfer parameters that modifies intermediate features maps in the image generator. Although images generated by NeuralHead accurately reenact head poses in driving images, they show significantly poorer performance in terms of identity preserving.

To summarise, X2Face ushers recent image warping-based methods into the self-supervised training paradigm, but this method can induce noticeable distortion in synthesized images as it directly warps the input images. MarioNETte decomposes sparse landmark points to guide the reenactment process, however, its PCA-based decomposition is a simplified modification of 3DMM, and this method requires multiple images as the input for best performance. In contrast, MeshGCN uses the full 3DMM to estimate optical flows, which subsequently brings more computation overhead. Lastly, NeuralHead leverages the raw landmark points in images to synthesize images. Although this method has lower head pose error, the generated faces share less similarity with the faces in source images.

Our method also leverages landmark coordinates to guide the reenactment process. When the source and the driving images share the same identity, we do not modify landmark coordinates, which is akin to NeuralHead. This is because in this scenario, the landmark coordinates in the driving images are the groundtruths, any modification would be redundant. When reenacting different people’s faces, we propose a conditional GAN to estimate reenacted landmark coordinates based on the face’s identity, the desired expression and head pose. Our method is efficient as it works on sparse 2D points. In addition, the training data can be conveniently annotated with the help of existing facial analysis tools such as OpenFace [24].

Table 2.1 shows the comparison of landmark coordinate reenactment between MeshGCN, MarioNETte and the proposed method. Both MeshGCN and MarioNETte need to estimate 3D landmark points. MeshGCN leverages dense points of 3D face models thus it requires a 3DMM face alignment model to regress 3DMM parameters from input images. MeshGCN directly estimates the optical flow by feeding landmark coordinates and corresponding adjacency matrix to a graph neural network. In comparison, MarioNETte only takes 68 points, it needs a 3D landmark detector to provide 3D coordinates. As mentioned above, coordinates estimated by MarioNETte are used to synthesize rasterized landmark images and feed to

the neural network for optical flow estimation. Our proposed landmark conditional GAN requires the information on facial action units that appears in the driving image and the angle of the driving face. In addition, sparse 2D landmark points are sufficient for our face reenactment method. Compared to MeshGCN, our method is more computationally efficient. Moreover, we do not make the assumption on the composition of landmark points or expression basis, thus our method has less inductive bias compared to MarioNETte. Evaluation results show that our method has lower head pose error and is capable of generating realistic images that preserve the source’s identity.

Table 2.1: Comparison of Landmark Coordinate Reenactment Methods

Model	Dimension	Usage of Coordinates	Required Extra Model
MeshGCN [15]	$\mathbb{R}^{53215 \times 3}$	estimate optical flow	3DMM Face Alignment
MarioNETte [16]	$\mathbb{R}^{68 \times 3}$	synthesize landmark image to estimate optical flow	3D Landmark Detection
Ours	$\mathbb{R}^{68 \times 2}$	guide landmark alignment and style transfer	AU Recognition and Head Pose Estimation

## 2.2 Generative Adversarial Networks

### 2.2.1 Vanilla GAN and Conditional GAN

Generative adversarial networks (GANs) are a family of neural networks that learn to map input from certain distribution to a desired distribution. A GAN consists of a generator  $G$  and a discriminator  $D$ , these two networks play a min-max game with a value function  $V(G, D)$  [25],

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

where  $\mathbf{x}$  is the real data sample,  $p_{\mathbf{x}}$  is the groundtruth distribution of the real data,  $\mathbf{z}$  is a random input and  $p(\mathbf{z})$  is the predefined distribution where  $\mathbf{z}$  is sampled from. Minimizing  $V(G, D)$  with respect to  $G$  implies that the log probability of  $D(G(\mathbf{z}))$  in Equation 2.1 needs to be maximized, namely the generator needs to produce realist samples such that the discriminator can be fooled. On the other hand, maximizing the value function with respect to  $D$  suggests that the discriminator should maximize  $D(\mathbf{x})$  while minimize  $D(G(\mathbf{z}))$ , which is equivalent to assigning high probability to real data samples and low probability to samples generated by  $G$ .

As shown in Equation 2.1, the input to the original GAN is a random vector sampled from a known distribution, such as a Gaussian distribution. The authors of conditional GAN [26] extended the framework of GAN by providing a conditional input  $y$  to both the generator and the discriminator, then  $D(\mathbf{x})$  and  $G(\mathbf{z})$  in the above equation becomes  $D(\mathbf{x}|y)$  and  $G(\mathbf{z}|y)$ , respectively, which means both the discriminator and the generator are now conditioned on the given  $y$ . Introducing the condition  $y$  to GAN brings more flexibility and control over the output of the generator, as the generator now takes the given condition into consideration. In terms of generating images with a conditional GAN, the condition  $y$  can be categorical labels [27], texts [28] or even images [29], depending on the objective of each task.

### 2.2.2 Image-to-Image Translation and Video Generation

Face reenactment is similar to image-to-image translation from the perspective of image generation. The objective of image-to-image translation is to transfer an image from one domain to another domain. For instance, the generator of Pix2Pix [29] takes semantic segmentation maps as input and synthesizes realistic RGB images. Regarding face reenactment, the image generator takes the source and the driving image as input, generating an image with the driving expression and head pose transferred to the source’s face. The relation between face reenactment and image-to-image translation enables that lessons learned from the latter can be applied to the former. Specifically, authors of Pix2Pix find that in order to synthesize clearer images, both the L1 loss on pixel values and the adversarial loss of GAN need to be used to supervise the model. The finding is also applied to recent self-supervised face reenactment methods, driving images are responsible for the L1 loss while a discriminator helps the generator synthesize more realistic images.

Vid2Vid [30] is another GAN that is closely related to face reenactment. The objective of Vid2Vid is to generate video frames by estimating optical flow. The authors of Vid2Vid formulate video frame generation as a conditional generation task, that is, given a set of previous video frames and semantic maps, the generator is expected to synthesize an RGB video frame for current time step. Unlike Pix2Pix, RGB images are not the only output of Vid2Vid’s generator. Vid2Vid also estimates an optical flow that warps the previous video frame. The final output of Vid2Vid is a composition of the warped video frame and a video frame synthesized by the generator. The process of estimating an optical flow and then warping the image to yield the final outcome shares many similarities with face reenactment. As discussed in Chapter 2.1, the general idea of image warping based methods [2, 15–19] is to estimate an optical flow based on the source and the driving image, then the source image

(or its feature maps) is warped to generate the reenacted face. For instance, the authors of [17,18] estimated key points from input images, their model would then predict an optical flow based these key points to warp the feature maps of input images. The authors of [15,16] estimated the optical flow from landmark feature maps and 3D meshes respectively, whereas the estimated optical flow would also be used to warp the feature maps of the source image.

GAN feature matching loss [31] proposed by the authors of Vid2Vid is another great contribution to face reenactment. This loss can be seen in [15,19], and [15] is the present state-of-the-art face reenactment method. GAN feature matching loss forces the synthesized images' features in the discriminator to be identical to their corresponding groundtruths images, providing a more direct feedback compared to the min-max loss in Equation 2.1. The use of GAN feature matching loss improves the speed of GAN training convergence and stability.

To summarise, GAN has undoubtedly become an indispensable component in face reenactment methods. Certain strategies proposed for training image generation GANs are also beneficial for face reenactment. In terms of our method, we introduce a GAN to estimate landmark coordinates, and our image generator is also trained with the GAN formulation. Details of GAN in our method are given in Chapter 3.

## 2.3 Vision Transformer

The original transformer [32] is an attention-based neural network designed to learn from sequential data for natural language processing. The key operation in the transformer architecture is the self-attention mechanism, which enables transformers to actively attend to all elements in the sequence, thus significantly boosting performance in a range of natural language processing tasks. Nowadays the state-of-the-art language models such as BERT [33] and GPT-3 [34] are all based on transformer.

The authors of Vision Transformer [35] brought the benefit of the original transformer to computer vision tasks. The most noticeable difference between Vision Transformer and the original transformer is how Vision Transformers process the input. Since transformer is designed to work on sequential data, the authors of Vision Transformer evenly crop input images into small image patches, these patches are projected into embedding vectors, then embedded vectors are stacked together to form the input sequence to the Vision Transformer. Details of Vision Transformer is introduced in Chapter 3.1. Unlike convolutional neural networks (CNNs) that are characterised by weight sharing and locality, Vision Transformer has less inductive bias [36], attention weights are dynamically computed depending on the

input and features are aggregated from all elements in the input sequence instead of an neighbouring area. This feature inspired us to apply Vision Transformer to optical flow estimation in our baseline method, as the model can directly aggregate features from all face regions to make the estimation. Many Vision Transformer variants [37–39] has been proposed since the publication of [35], these methods focus on introduce beneficial properties of CNNs into the architecture designs of Vision Transformers, such as transforming sequential features into 2D to leverage the locality of features. To verify the above idea that the attention over the entire input is beneficial for optical flow estimation, our baseline model keep the original design of Vision Transformer instead of using its more recent variants. Experiments in [35] show that Vision Transformer would require much more parameters and tens of millions training examples to reach the same level of performance as CNNs, we show that a shallow Vision Transformer is also a good optical flow estimator for face reenactment. Details of how Vision Transformer operates in our method is given in Chapter 3.

## 2.4 Federated Learning

Federated learning is first proposed by Google to train machine learning models in the distributed setting, thus data leakage can be avoided [40]. In contrast to conventional deep learning paradigm, federated learning does not require all the data and models to be on the same machine. Multiple client models are trained in parallel without transferring local data to other machines. A global model is often required to aggregate client models and make predictions for the desired task. The neural network architectures in federated learning are not much different from that of the centralized learning, however, research in federated learning faces unique challenges. The statistical heterogeneity in data is one of the most outstanding issues of federated learning. Since data is now owned by different clients, data from different sources may no longer follow the same distribution. For instance, images of clothing can vary widely around the world [41]. Statistical heterogeneity in data violates the assumption in conventional machine learning that all the training data are independent and identically distributed, client models trained on such data may have different or even contradicting views on the task in question. Therefore, research on aggregation strategies in federated learning is still an active area.

Federated Averaging (FedAvg) [42] is a pioneering method for aggregating client models in federated learning. Client models  $c_i$  are first initialized with identical parameters as the global model  $x$ . In each round of the global model update, a few client models are random selected and fed with local data to conduct local update. When selected clients are updated,

the global model aggregate the weights of client models in the following way,

$$w_x = \sum_{i=1}^K \frac{n_i}{n} w_i \quad (2.2)$$

where  $w_x$  is the weights of the global model,  $K$  is the number of randomly selected client models,  $w_i$  is the weight of a chosen client model  $i$ ,  $n$  is the total number of training samples combined, and  $n_i$  is the number of training samples on the client machine  $i$ . Authors of FedAvg assumed that randomly selecting client models in each training round is analogous to the dropout operation in neural networks, acting as a regularization while allowing for faster training [43].

Regarding federated learning for image generation, there are relatively few literature on this topic, recent research focuses more on image classification to validate the performance of proposed algorithms. The work of FedGAN [44] extended FedAvg to the realm of image generation through GAN. FedGAN is very similar to the strategy of FedAvg, it distributes client generators and client discriminators, and aggregates the global generator and global discriminator through FedAvg. In this paper, we take the same approach as FedGAN to evaluate our method in the federated learning setting. Details of FedGAN in our face reenactment method is introduced in Chapter 4.

## 2.5 Summary

In Chapter 2.1.1, we described the general pipeline of rendering-based face reenactment methods and showed typical examples of this type. These methods often require a large number of input, making it difficult to train and apply such methods, recent research focuses on image warping-based methods for better solution.

In Chapter 2.1.2, we explained the pipeline and self-supervised strategy of X2Face [2]. The strategy proposed by authors of X2Face greatly evolved the training of face reenactment models and gave rise to a range of image warping-based methods, among which we emphasizes two state-of-the-art methods, MeshGCN [15] and MarioNETte [16]. We discussed in detail how these methods reenact landmark coordinates to achieve more accurate reenactment. We are convinced that landmark coordinates are rather useful for face reenactment and we argue if the landmark coordinates are reenacted with high precision, we can more directly leverage them to guide the reenactment process. We compared our landmark reenactment method with MeshGCN and MarioNet in Table 2.1.

In Chapter 2.2.1, we showed the definition of the vanilla generative adversarial networks

(GAN) and its conditional variant. In Chapter 2.2.2, we reviewed how researchers extended the framework of GAN to two image generation tasks: image-to-image translation and video generation. Research on these two tasks greatly benefit the development of face reenactment, because findings on how to generate realistic images in these tasks are also applicable to face reenactment and have been verified by existing research [15,16].

In Chapter 2.3, we introduced Vision Transformer. The original transformer is a neural network designed for natural language processing. The self-attention module in a transform grants the model exceptional capability to dynamically process input data. Authors of Vision Transformer adapted this network to computer vision tasks and Vision Transformer has outperformed classic convolutional neural networks such as ResNet. We intend to bring Vision Transformer to our method because we believe the attention mechanism in Vision Transformer is also beneficial for face reenactment.

In Chapter 2.4, we briefly reviewed federated learning and explained the FedAvg algorithm, the indisputable foundation of federated learning. However, FedAvg is not designed for generative tasks, we therefore resort to FedGAN, an extension of FedAvg to conduct experiments on federated face reenactment.



### 3 Face Reenactment with Generative Landmark Guidance

In this chapter, we present our face reenactment method. We are motivated by the intuition that accurate landmark coordinates may lead to accurate reenactment. In addition to estimating optical flow that warps the input’s feature maps, we individually reenact the eyes, nose and the mouth and align them with desired landmark coordinates to guide our image generator.

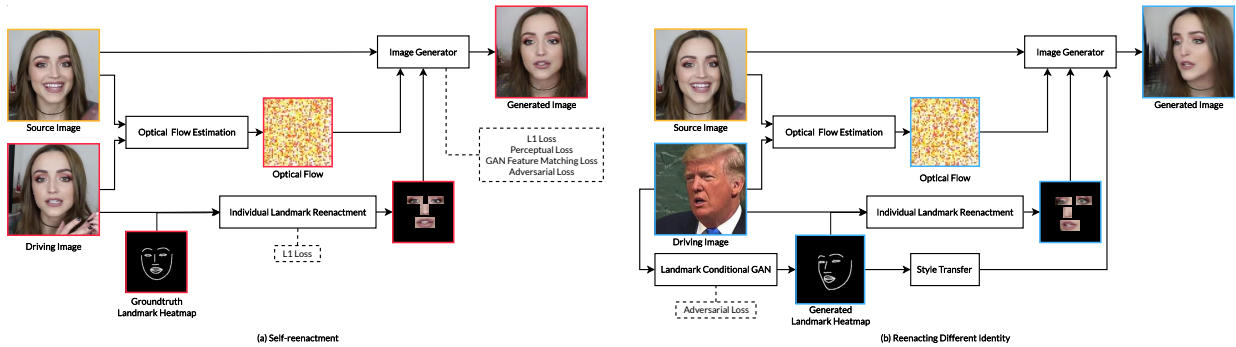


Figure 3.1: Overview of proposed method for self-reenactment and reenacting different identities. Dashed boxes show loss functions that are responsible for the corresponding module.

If the source and driving images share the same identity, landmark coordinates in the driving images provide the groundtruth-level coordinates for alignment. However, when the source and driving images have different identities, aligning landmarks with coordinates in the driving image can cause severe identity preserving problems discussed in Chapter 1. This is why we need to find a way to estimate landmark coordinates for this situation. Existing methods take the 3DMM formulation to reenact landmark coordinates, we instead consider two crucial aspects of face reenactment: head poses and facial expressions. Head poses can be quantified by the rotation angles while facial expressions can be constructed with the help of facial action units. We therefore propose a GAN conditioned on the driving image’s head pose angles and facial action units to estimate landmark coordinates.

Figure 3.1 shows the overall framework of our face reenactment model. In general, we first estimate an optical flow based on input images. Then the eyes, nose and mouth in the source image are individually reenacted and aligned with corresponding landmark coordinates. Lastly, we warp the feature maps of the source image with estimated optical flow, and use aligned landmarks to guide the subsequent image generation process. Figure 3.1.

(a) shows the self-reenactment scenario, in which landmark coordinates does not need to be estimated. Figure 3.1. (b) shows the process of reenacting different identities. Landmark coordinates are estimated by the proposed conditional GAN and we further add a style transfer branch to improve the realism of generated faces.

### 3.1 Optical Flow Estimation

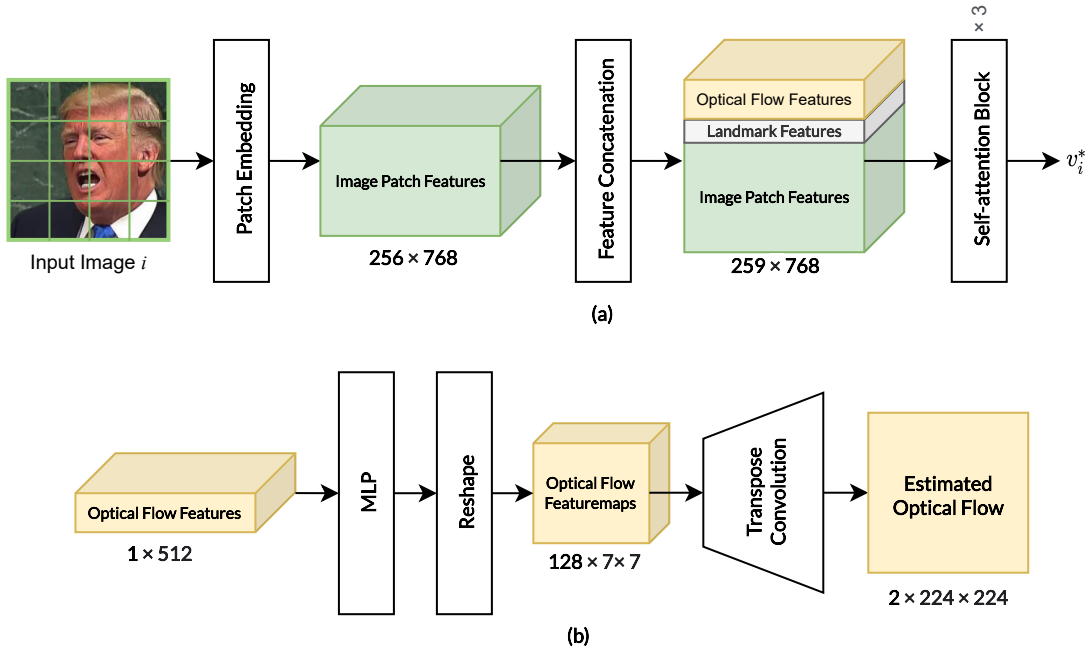


Figure 3.2: Architecture of Facial Feature Extraction Module. (a) Extracting features from input images; (b) Estimating the optical flow based on extracted features.

Facial feature extractor is responsible for extracting and aggregate image features for optical flow estimation. The extractor is comprised of a Vision Transformer with three layers. The architecture of this module is shown in Figure 3.1. An input image  $i$  with size  $224 \times 224$  is evenly divided into 256 patches with size  $14 \times 14$ . Each image patch is flattened into a  $1 \times 196$  vector, then embedded into a  $1 \times 768$  vector through a fully-connected layer, resulting in a  $256 \times 768$  tensor  $v_i$  for the input image. In addition, a tensor  $t \in \mathbb{R}^{3 \times 768}$  with learn-able initial values are concatenated to  $v_i$ , the first two rows of  $t$  store features for the optical flow estimation, and the third row of  $t$  contains features for landmark coordinate regression, which acts as an auxiliary task that helps the model perceive human faces. After

an input image being embedded into  $v_i \in \mathbb{R}^{259 \times 768}$ , it further goes through three self-attention layers. The self-attention process is given as follows.

$$Q = v_i W_q, K = v_i W_k, V = v_i W_v \quad (3.1)$$

$$\alpha = \text{softmax}(QK^T / \sqrt{d_k}), v_i^* = \alpha V \quad (3.2)$$

where  $W_q \in \mathbb{R}^{768 \times d_q}$ ,  $W_k \in \mathbb{R}^{768 \times d_k}$  and  $W_v \in \mathbb{R}^{768 \times d_v}$  are learn-able parameters,  $d_q = d_k = d_v = 768$ ,  $\alpha \in \mathbb{R}^{259 \times 259}$  is the attention score given the input tensor  $v_i$ , and  $v_i^* \in \mathbb{R}^{259 \times 768}$  is the output of the self-attention operation, it further goes through a multi-layer perceptron (MLP) to yield the final result of a transformer block. In Equation 3.1,  $Q$ ,  $K$  and  $V$  respectively stands for the query, key and value matrix. The intuition behind this formulation is that  $Q$  raises queries about the input features,  $K$  holds the addresses to retrievable information, while  $V$  stores the information to be retrieved. By doing a multiplication between  $Q$  and  $K$  in Equation 3.2, the attention score  $\alpha$  indicates the relevance of each feature in  $V$  regarding the query  $Q$ . Features in  $V$  are then aggregated based on the estimated attention score.

Optical flow features for the source and the driving image are denoted by  $u_s, u_d \in \mathbb{R}^{2 \times 768}$  respectively.  $u_s$  and  $u_d$  are first compressed to  $\mathbb{R}^{2 \times 128}$  then reshaped to  $\mathbb{R}^{1 \times 256}$ . As shown in Figure 3.2 (b), these two features are concatenated to obtain a feature of shape  $\mathbb{R}^{1 \times 512}$  and sent to an MLP, resulting in  $f \in \mathbb{R}^{1 \times 6272}$ ,  $f$  is reshaped to  $\mathbb{R}^{7 \times 7 \times 128}$  and after going through a series of transpose convolutional layers, the estimated optical flow  $f^* \in \mathbb{R}^{2 \times 224 \times 224}$  is obtained.

## 3.2 Individual Landmark Reenactment

To explicitly guide the reenactment process with landmark locations, the eyes, the nose, and the mouth of the source are individually reenacted and placed at the same location as their corresponding landmark coordinates. We use four convolutional neural networks with an identical architecture, and each of them is dedicated to reenacting a different part of the face, namely the left eye, the right eye, the nose, and the mouth. Figure 3.3(a) shows we concurrently reenact selected landmarks; Figure 3.3(b) gives an example of the crop of the mouth from the source image, along with its counterpart from the landmark heatmap of the driving image are first sent to convolution layers, with the size of feature maps reduced by max pooling, then feature maps of the RGB mouth crop and that of

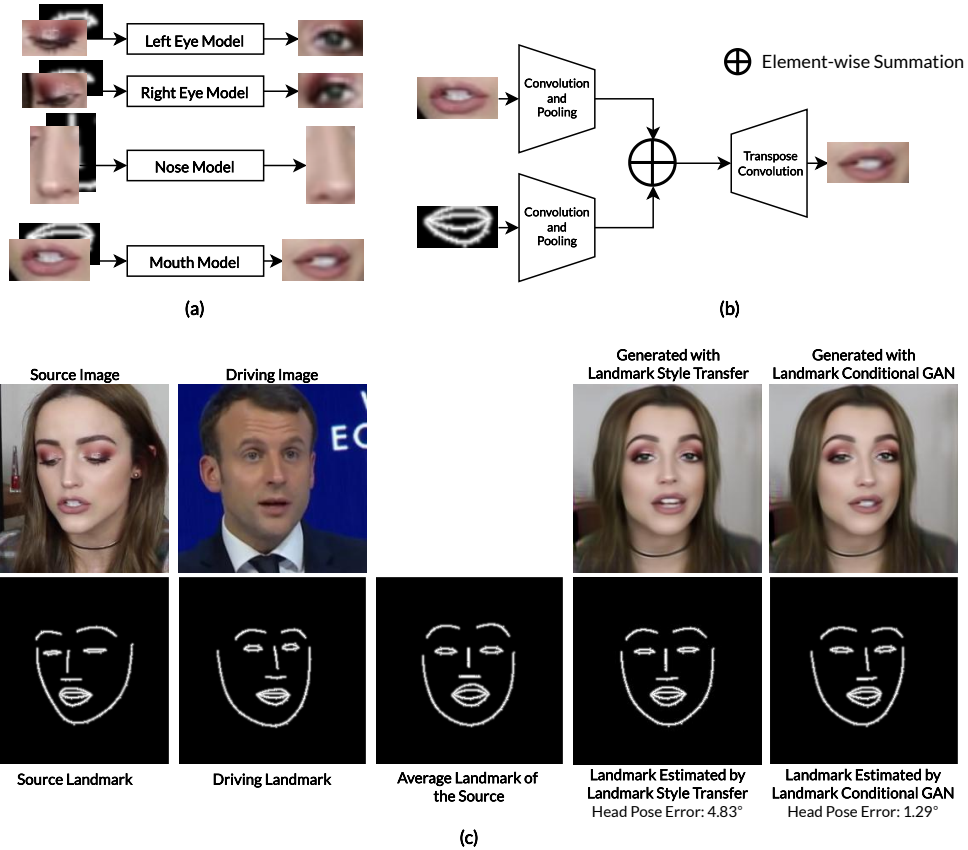


Figure 3.3: Individual landmark reenactment and examples of landmark generation.

(a) Individual landmarks are concurrently reenacted with models that share the same architecture. (b) Example of mouth reenactment showing the architecture of the landmark reenactment model. (c) Landmark coordinates estimated by style transfer significantly sacrifice head pose accuracy for identity preserving, whereas landmark conditional GAN better balances this trade off.

the landmark heatmap are added element-wise and sent to transpose convolution layers to generate reenacted landmarks. All crops are fixed-sized and they are cropped around the centre point of corresponding landmark coordinates. The size of a landmark crop takes the value of the average size of corresponding landmark in the dataset. The landmark heatmap is obtained by first drawing 68 facial landmark points on a  $224 \times 224$  image with black background, then points are connected by fitting B-spline curves, drawing the outlines of the face, eyes, eye brows, nose and mouth. When all landmarks are reenacted, they are directly placed on another blank  $224 \times 224$  image  $I_p$ , and their centre point all align with the centre point of corresponding parts in the landmark heatmap.

### 3.3 Landmark Estimation

Although our landmark reenactment module relies on the face sketch generated by driving landmark coordinates, no modification on landmark coordinates is needed during training as source images and driving images share the same identity. When we reenact faces with different identities, this leads to the identity preserving problems described in Chapter 1 due to the identity mismatch between the source image and the driving sketch.

#### 3.3.1 Landmark Style Transfer

To remedy this, we modify driving landmark coordinates by treating it as a style transfer problem. Inspired by [45], to adapt the driving person’s landmark coordinates to the landmark style of the source person, we align the mean and variance of the driving coordinates  $L_{driving}$  with those of the source coordinates  $L_{source}$ ,

$$L_{reen} = \frac{L_{driving} - \mu_{driving}}{\sigma_{driving}} \times \sigma_{source} + \mu_{source} \quad (3.3)$$

$\mu_{source}, \sigma_{source}, \mu_{driving}, \sigma_{driving}$  can be obtained by computing the mean and variance of each person’s landmark coordinates in the dataset, and  $L_{reen}$  is derived from modulating  $L_{driving}$  with the computed statistics. We also shift  $L_{reenact}$  such that its centre point is at the same location as  $L_{driving}$ . Figure 3.3(b) shows an example the driving face sketch generated by the original landmark coordinates and the one generated by style-transferred coordinates.

#### 3.3.2 Landmark Conditional GAN

One major problem with the above landmark style transfer is that Equation 3.3 pushes landmark coordinates towards the average head pose in the dataset instead of truthfully acting as the desired pose. As shown in Figure 3.3(b), landmark coordinates modified by style transfer To remedy this problem, we propose the landmark conditional GAN as a more reliable estimator.

The input to our conditional GAN is inspired by the evaluation metrics of face reenactment methods, specifically, we feed the source’s identity, the driving’s head pose, and facial action units [46] appeared on the driving’s face into the generator to obtain 68 2-D landmark coordinates. Facial action units(AUs) are predefined basic muscle movements on human faces. Figure 3.4 shows selected AUs in our method, these AUs are also used for face reenactment evaluation.


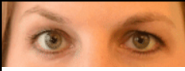








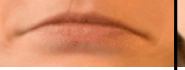







AU1	AU2	AU4	AU5	AU6	AU7
					
Inner Brow Raiser	Outer Brow Raiser	Brow Lowerer	Upper Lid Raiser	Cheek Raiser	Lid Tightener
AU9	AU10	AU12	AU14	AU15	AU17
					
Nose Wrinkler	Upper Lip Raiser	Lip Corner Puller	Dimpler	Lip Corner Depressor	Chin Raiser
AU20	AU23	AU25	AU26	AU28	AU45
					
Lip Stretcher	Lip Tightener	Lips Apart	Jaw Drop	Lip Suck	Blink

Figure 3.4: Facial action units (AUs) for estimating landmarks and evaluating performance.

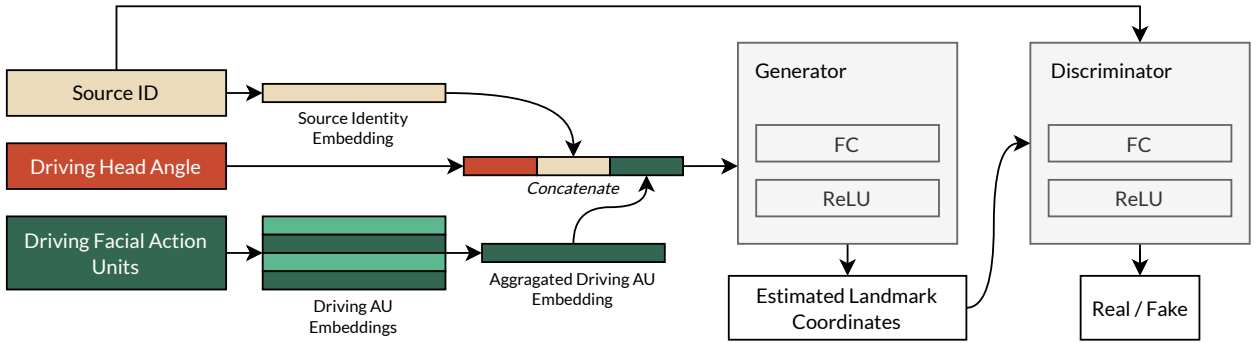


Figure 3.5: The architecture of landmark coordinate conditional GAN.

The convention of AU study is that a complex expression can be expressed by the addition of many different facial action units. For instance, an unhappy mouth can be expressed as AU15+AU17. We then took a similar approach to process AUs in the conditional GAN. Embedding vectors of facial action units appeared in the driving image are first selected, then these vectors are summed up to yield the overall expression feature for the input. Proposed landmark conditional GAN can be formulated as,

$$L_{reen} = g(\text{ID}_s, (\alpha, \beta, \gamma), \sum_{i=1}^{18} \mathbb{1}AU_i) \quad (3.4)$$

where  $g(\cdot)$  is the generator of landmark conditional GAN,  $\text{ID}_s$  is the identity of the source

image,  $\alpha$ ,  $\beta$ , and  $\gamma$  are normalized angles of the driving’s head pose, the value  $\mathbb{1}$  is 1 if an AU appears in the driving image, otherwise it is 0,  $AU_i$  is the embedding vector of the  $i$ -th AU. We consider 18 AUs shown in Figure 3.4 as these AUs are commonly used for face reenactment evaluation. The overall architecture of landmark conditional GAN is shown in Figure 3.5.

### 3.4 Image Generator

The face reenactment module is a U-Net-like convolutional neural network with one intermediate skip-connection. Figure 3.6 shows its overall architecture. The source image is first sent to three convolutional layers with the size of its feature map  $r$  being reduced to  $58 \times 58$ , then the estimated optical flow map  $f^*$  (Chapter 3.1) with size  $224 \times 224$  is resized to match the size of  $r$  and warps  $r$ , yielding the warped feature map  $r^*$ . The image  $I_p$  with reenacted landmark parts from the landmark reenactment module (Chapter 3.2) is also resized to  $58 \times 58$  and concatenated to  $r^*$ . The concatenated feature map  $r_{cat}^*$  continues to go through intermediate convolutional layers with no change in feature map size, then  $r^*$  is concatenated to  $r_{cat}^*$  through the skip connection, the resulting feature map is further upsampled through bilinear interpolation and processed by convolution layers to generate the final reenacted image. The use of bilinear upsampling is aiming for alleviating the checkerboard artifact in images generated by convolutional neural networks [47].

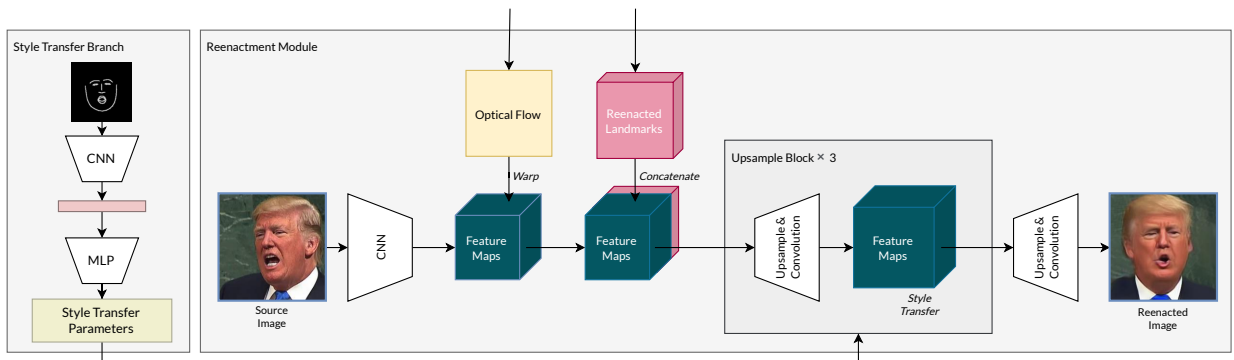


Figure 3.6: The style transfer branch and the image generator.

In the case of reenacting different identities, although our landmark estimation methods greatly alleviated the identity preserving problem, unnatural face deformations exist as a result of inaccurate optical flow estimation. To rectify this issue, we further introduce a style transfer branch to the generator. The architecture of the style transfer branch is inspired by StyleGAN2 [48]. Instead of estimating style transfer parameters from random

inputs, our model takes 1-channel landmark heatmaps as input. These landmark heatmaps are generated by first estimating the landmark coordinates using the conditional GAN in Chapter 3.3.2, then b-spline curves are fitted between adjacent landmark points that belong to the same landmark part, namely drawing out the contours of the face, eyes, eyebrows, nose, and mouth. The use of heatmaps avoids the identity leak which is destined to happen if RGB driving images were used. Furthermore, since the heatmaps are generated based on coordinates estimated by our landmark conditional GAN, the identity information of the driving person is excluded as much as possible. The architecture of the style transfer branch is shown in Figure 3.5.

### 3.5 Loss Function

Overall we use the weighted sum of four types of loss function to train our face reenactment model, namely,

$$L = \lambda_l L_1 + \lambda_g L_{Adv} + \lambda_f L_{FM} + \lambda_p L_P \quad (3.5)$$

Each term in Equation 3.5 are defined as follows, the loss weights  $\lambda$  are chosen based on our observation during experiments.

- **L1 Loss:** L1 loss is responsible for supervising the pixel values in generated images. During training, driving images  $I_d$  are also the groundtruths for generated images  $I_g$ , L1 loss is computed between these images through Equation 3.6. The weight  $\lambda_l$  on this loss is set to 20 for the entire image, and 5 for individually reenacted landmarks.  $H$  and  $W$  are the height and the width of the image respectively.  $I^{ij}$  stands for the pixel value at  $(i, j)$  in an image. We find that putting more weight on the L1 loss prevents the model from generating unexpected artifacts.

$$L_1 = \frac{1}{HW} \sum_{i=1}^H \sum_{j=1}^W |I_d^{ij} - I_g^{ij}| \quad (3.6)$$

- **Adversarial Loss:** The adversarial loss we used for training is the same as [49]. Driving images are treated as "real samples" while reenacted images are labeled as "fake". We set the weight  $\lambda_g$  for this loss to 1.

$$L_{Adv} = [\mathbb{E}_{I_d \sim p_{I_d}} [\log D(I_d)] + \mathbb{E}_{I_g \sim p_{I_g}} [\log(1 - D(I_g))]] \quad (3.7)$$



- GAN Feature Matching Loss [31]: GAN feature matching loss requires the discriminator to return intermediate features of real and generative samples, forcing these features to be the same. In Equation 3.8,  $K$  is the number of neural network layers in the discriminator,  $D^k(I)$  denotes the feature maps extracted from the  $k$ -th layer of the discriminator given an input  $I$ , and  $\|\cdot\|_1$  stands for the L1 loss. The weight  $\lambda_f$  for this loss is also set to 1. For generative tasks with groundtruth samples, GAN feature matching loss makes the training more stable and converge faster.

$$L_{FM} = \sum_{k=1}^K \|D^k(I_d) - D^k(I_g)\|_1 \quad (3.8)$$

- Perceptual Loss [50]: Perceptual loss relies on a pretrained VGG model to extract shallow visual features for real and generative samples. Pushing these features to be close ensures that low level features in the generated image, such as the shape of the face and shoulder, to be more realistic. In Equation 3.9, we choose the first 30 layers of the VGG model pretrained on the ImageNet dataset. These layers are divided into the following 5 groups,  $\{(1, 2), (3, 7), (8, 12), (13, 21), (22, 30)\}$ , where each group is denoted by the sequential orders of its starting and ending layers. For instance, the second group (3, 7) starts at the 3rd layer in the VGG model and ends with the 7th layer of the VGG model.  $J$  is the number of groups and in this case we have  $J = 5$ .  $V^j(I)$  stands for the output feature maps from layers of the  $j$ -th group given an input  $I$ . The weight  $\lambda_p$  for this loss is set to 10.

$$L_P = \sum_{j=1}^J \|V^j(I_d) - V^j(I_g)\|_1 \quad (3.9)$$

### 3.6 Application: Federated Face Reenactment

In previous sections of this chapter, we present the pipeline of proposed face reenactment method, in this section we show how we apply it to federated learning. Currently there are only a few studies on generative tasks in federated learning and little research has been conducted on federated face reenactment. We therefore adapt the CelebV dataset and apply the FedGAN algorithm, which is the generative version of the classic FedAvg algorithm.

### 3.6.1 Federated Learning Configurations

Figure 3.7 shows the federated learning configuration for our face reenactment method. To facilitate federated learning with existing face reenactment datasets, we evenly assign images of a specific identity in the CelebV dataset [7] to one of the clients, each client can only access images of appointed identity. We assume that face images of different people are sampled from different distributions, thus, dividing datasets based on people’s identities simulates the statistical heterogeneity in federated learning.

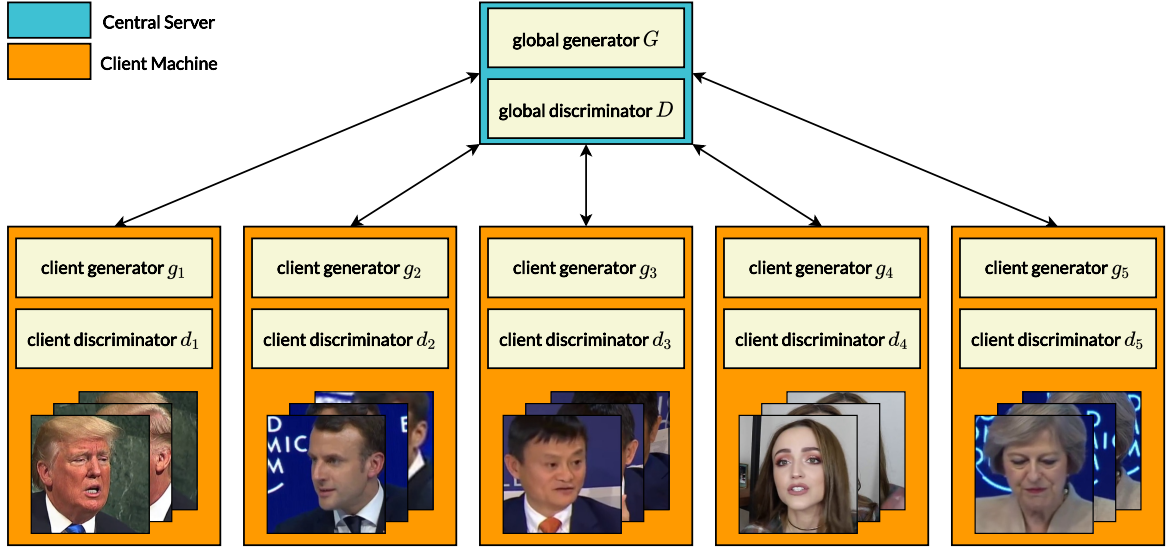


Figure 3.7: Configuration of Federated Face Reenactment.

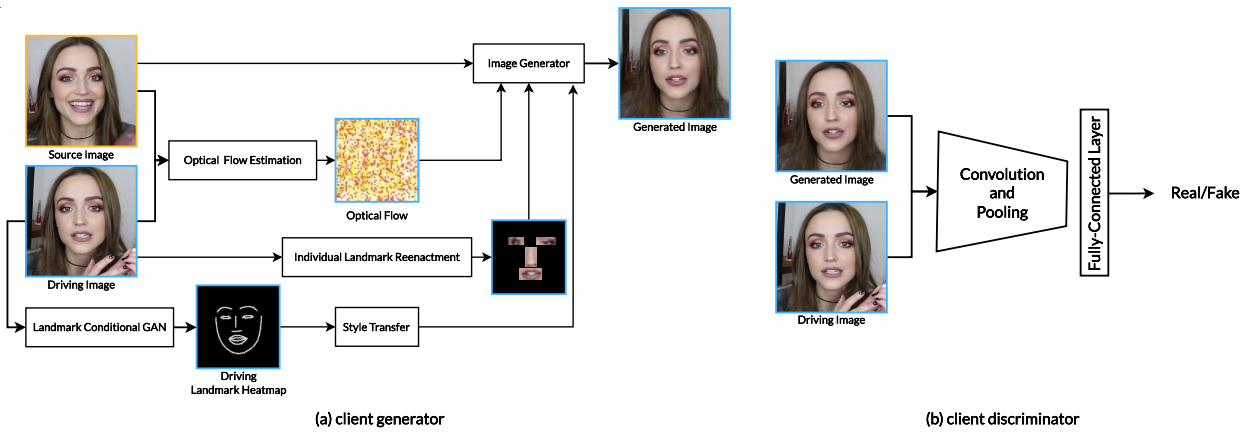


Figure 3.8: Architecture of Client Models.

Given a client  $c_i$ , it maintains the local training of two models, the local generator  $g_i$  and local discriminator  $d_i$ . For a client generator  $g_i$ . Figure 3.8 shows the architecture of client

models. The architecture of the client generator is identical to the model in Figure 3.1 (b), the only difference is that the model can now only access images of one specific person as federated learning forbids local clients to access remote data. The global generator and the global discriminator share the same architecture as their client counterparts. Notice that clients do not communicate between each other, they send locally updated models to and receive aggregated models from the central server. Lastly, loss functions are the same as the ones discussed in Section 3.

### 3.6.2 Model Aggregation

We directly apply FedGAN [44] shown in Algorithm 1 for our federated face reenactment. Hyperparameters such as the learning rate and the termination threshold are chosen based on our experience. FedGAN is the an adaptation of FedAvg [42] for GAN. It deploys a generator and a discriminator on each client, then update these models the same way as FedAvg, namely aggregating client models by taking the average of their weights.

---

**Algorithm 1:** Federated Face Reenactment.  $L_G$  is the generator loss,  $L_D$  is the discriminator loss,  $b$  is a mini-batch of local data.

---

**Initialization:** Initialize the global generator and discriminator  $G, D$  with parameters  $\omega_G$  and  $\theta_D$  respectively. Set the learning rate  $\eta_g$  for client generators and  $\eta_d$  for discriminators as 0.0001. Set the threshold  $\tau$  as 3, and the local update step  $E$  as 100.

```

1 while  $round \leq 5$  or  $Var(L_G) \geq \tau$  do
2   for client  $i \in \{1, 2, \dots, 5\}$  in parallel do
3      $\omega_i \leftarrow \omega_G$  ; // synchronize with global generator
4      $\theta_i \leftarrow \theta_D$  ; // synchronize with global discriminator
5     for local update step  $e = 1, 2, \dots, E$  do
6        $\omega_i = \omega_i - \eta_g \nabla L_G(\omega_i; b)$  ; // update local generator
7        $\theta_i = \theta_i - \eta_d \nabla L_D(\theta_i; b)$  ; // update local discriminator
8      $\omega_G = \sum_{i=1}^5 \frac{1}{5} \omega_i$  ; // aggregate global generator
9      $\theta_D = \sum_{i=1}^5 \frac{1}{5} \theta_i$  ; // aggregate global discriminator

```

---

Given the fact that we have only 5 clients, we do not have to randomly choose clients to speed up the training process, therefore no client model is left out when the global model aggregates parameters from clients. Also, because the data is evenly distributed among clients, all client models share the same weight in the aggregation stage. Instead of training the model for predefined epochs, when the variance of the generator loss  $Var(L_G)$  from the

last 5 rounds is less than a given threshold  $\tau$ , the training is terminated and global models are assumed to be converged. Algorithm 1 shows the procedures of federated face reenactment discussed in this chapter.

### 3.7 Summary

In this chapter, we showed each module of our proposed method and loss functions we used to train our model. Our method has three major stages: optical flow estimation, individual landmark reenactment and image generation.

In the optical flow estimation stage, we feed the input source and driving image to a neural network to extract their respective features, and we combine extracted features to estimate an optical flow which operate on the feature maps of the source image.

When reenacting individual landmarks, we require the RGB landmarks cropped from the source image and corresponding crops from the landmark heatmap generated from landmark coordinates. Since it is necessary to reenact landmark coordinates when the source and the driving have different identities, we first propose a coordinate style transfer method, then we introduce a landmark conditional GAN to better estimate reenacted landmark coordinates. Reenacted landmarks are copied to a blank image and aligned with the driving landmark coordinates or estimated coordinates, depending on whether the source and the driving image share the same identity or not.

In the final stage of our method, we combine outputs from last two stages to synthesize reenacted face images. The source image is first fed to the image generator, then the estimated optical flow is used to warp the feature maps of the source image. We further concatenate reenacted landmarks to warped feature maps and use transpose convolution to generate the final output. Also, if the source and the driving images have different identity, we adopt a style transfer branch to help synthesize more realistic images.

We detail the configuration for federated face reenactment with the proposed method. We split the CelebV dataset based on people’s identities as we assume that face images of different people are sampled from different distributions. By splitting the dataset in this way, we are simulating the data heterogeneity in federated learning. We assign a client generator and a client discriminator to each client end, and we also maintain a global generator and global discriminator. We follow Algorithm 1 to update client model and aggregate parameters for the global model.

Experimental results on conventional and federated face reenactment are shown in Chapter 4. Ablation studies on proposed landmark style transfer, landmark conditional GAN and

the style transfer branch are also included in Chapter 4.

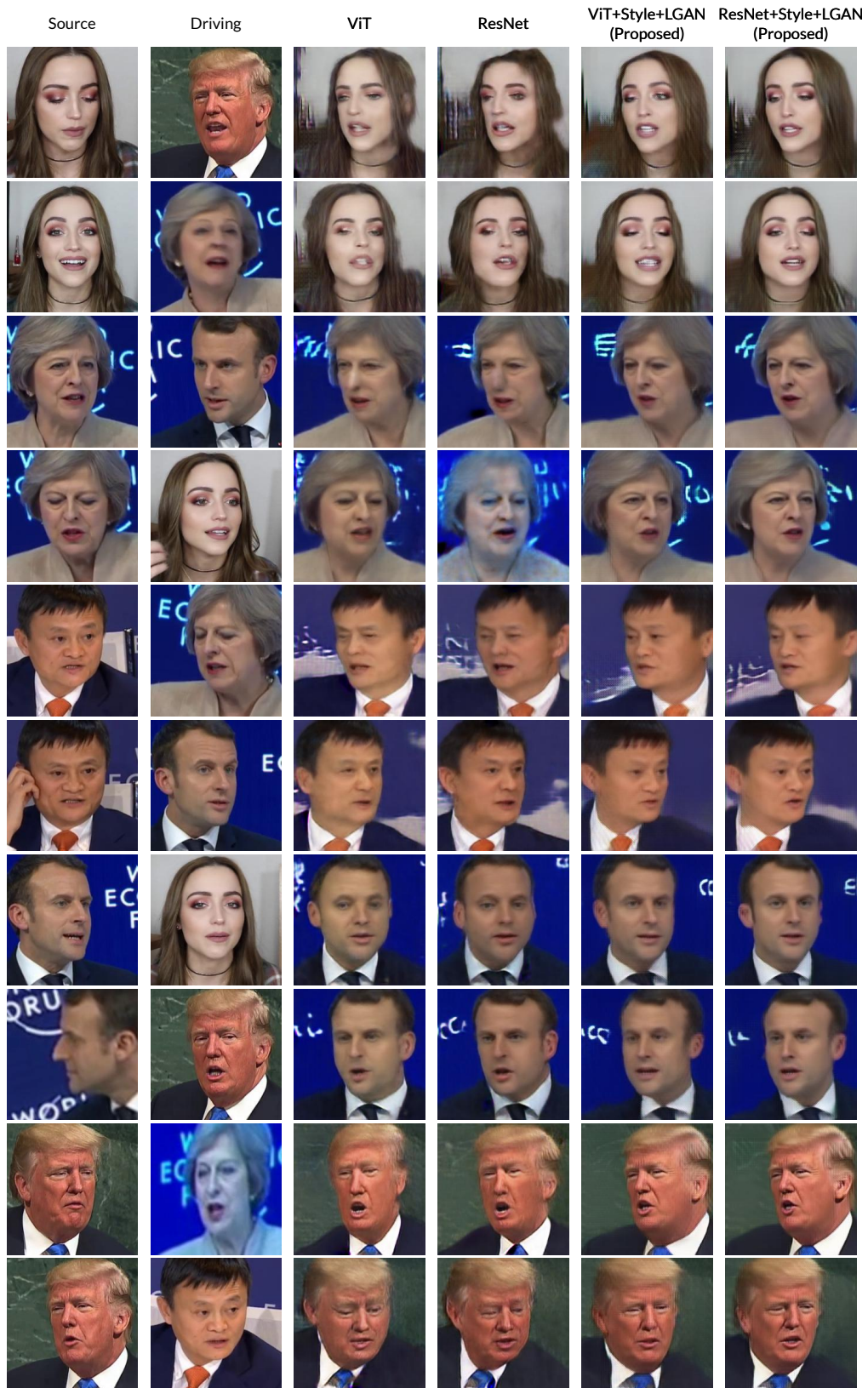


Figure 3.9: Qualitative results of proposed models on CelebV dataset.



Figure 3.10: Qualitative results of proposed models on CelebV dataset.

## 4 Experiments

### 4.1 Datasets and Experimental Settings

We evaluated our methods on the VoxCeleb1 dataset for self-reenactment, and the CelebV dataset for reenacting different identities and federated face reenactment. VoxCeleb1 is a dataset with 22,496 video clips extracted from YouTube. It contains 1,251 identities, and people’s faces have been cropped into  $256 \times 256$  images. CelebV has around 40,000 images for each of the five people in the dataset. For each person, their images are sampled from the same video and has also been cropped into  $256 \times 256$  images.

The training and test sets for evaluating our method are shown in Table 4.1. For self-reenactment, we follow the protocol in [15, 16] and trained our model on the VoxCeleb1 dataset. The test set for evaluation consists of 100 videos from the test split given by authors of the dataset, 2,083 source-driving image pairs are sampled from these videos for evaluation. For reenacting different identities, since our conditional landmark GAN and style transfer module require the information of known identities, we evaluated our method in two scenarios. The first scenario follows [15, 16] and aims at reenacting unseen identities. Models are only trained on the VoxCeleb1 dataset, however, the test set are image pairs sampled from the CelebV dataset. For each person in CelebV, 2,000 source-driving image pairs are randomly sampled. In the second scenario, models are only trained on the CelebV dataset. Test set in this case also comprises 2,000 source-driving image pairs randomly sampled for each person. For federated learning, the test set is the same as the above test sets on CelebV. The remaining data of CelebV is evenly divided based on the five identities in the dataset, then distributed to each client.

Table 4.1: Details of datasets for evaluated tasks.

Reenactment Task	Training Set	Test Set
self-reenactment	VoxCeleb	2,083 pairs (VoxCeleb)
different and unknown identities	VoxCeleb	10,000 pairs (CelebV)
different but known identities	CelebV	10,000 pairs (CelebV)
federated learning	CelebV	10,000 pairs (CelebV)

We evaluated two model variants, denoted by their backbone network for optical flow estimation, namely ViT and ResNet. Further ablation studies were also conducted to validate our proposed methods. The ViT model has three Vision Transformer layers for optical flow estimation, for the ResNet variant, transformer layers are replaced by ResNet-34. In the



work of [35], a modified ResNet-50 (25 million parameters) outperforms the base 12-layer Vision Transformer (86 million parameters) on ImageNet top-1 accuracy by 10% with a pre-training dataset of 10M images. Given that there are three Vision Transformer layers (19M parameters) in our baseline model, we hence choose ResNet-34 (21M parameters) for comparison, which is shallower than ResNet-50. Additionally, we applied landmark style transfer described in Chapter 3.3.1 to both models and evaluated their performance accordingly. Models with landmark style transfer are denoted by ViT+LSt and ResNet-34+LSt.

## 4.2 Metrics

Performance on self-reenactment was evaluated through the following metrics, cosine similarity (CSIM), structural similarity (SSIM) [51], peak signal-to-noise ratio (PSNR), root mean square error of head pose angles (PRMSE), and the ratio of correct facial action units (AUCON). CSIM measures the model’s capability on identity preserving. It is derived from the cosine similarity between embedding vectors of the source and generated images, these vectors are extracted by a pretrained face recognition model ArcFace [52], namely,

$$CSIM = \frac{ArcFace(I_s)ArcFace(I_g)}{\|ArcFace(I_s)\|\|ArcFace(I_g)\|} \quad (4.1)$$

where  $ArcFace(\cdot)$  yields the embedding vector of input face image,  $I_s$  is the source image, and  $I_g$  is the generated image.

SSIM and PSNR are exclusive to self-reenactment evaluation as they both require ground-truth images to derive, which is not possible for reenacting different identities. PSNR evaluates low-level similarity between generated images and ground-truths, and is defined as follows,

$$PSNR = 10 \log\left(\frac{255^2}{\frac{1}{HW} \sum_{i=0}^H \sum_{j=0}^W (I_d^{ij} - I_g^{ij})^2}\right) \quad (4.2)$$

where 255 is the maximum pixel value in an image,  $H$  and  $W$  are the height and the width of images respectively,  $I_d$  is the driving image,  $I^{ij}$  denotes the pixel value of image  $I$  at  $(i, j)$ .

SSIM jointly evaluates the contrast, luminance, and structural similarity between images in the following way,

$$SSIM = [l(I_d, I_g)]^\alpha \cdot [c(I_d, I_g)]^\beta \cdot [s(I_d, I_g)]^\gamma = \frac{(2\mu_d\mu_g + C_1)(2\sigma_{dg} + C_2)}{(\mu_d^2 + \mu_g^2 + C_1)(\sigma_d^2 + \sigma_g^2 + C_2)} \quad (4.3)$$

where  $l(\cdot)$ ,  $c(\cdot)$  and  $s(\cdot)$  respectively measure the luminance, contrast and structural similarity between input images. If  $\alpha$ ,  $\beta$  and  $\gamma$  are all set to 1, SSIM can then be simplified into the form on the rightmost side of Equation 4.3.  $\mu_d$  and  $\mu_g$  are the mean pixel values of the driving image and the generated image respectively,  $\sigma_d$  and  $\sigma_g$  are the standard deviations of corresponding images, and  $\sigma_{dg}$  is the correlation coefficient of pixel values between the driving and generated image. Both  $C_1$  and  $C_2$  are close-to-zero constants to ensure the computational stability.

Head pose angles and facial action units are detected by OpenFace [24]. PRMSE is computed by calculating the root mean square error of head pose angles of the generated image compared against those of the driving image. For AUCON, both the driving and generated image are sent to OpenFace, the returned results show if facial action units in Figure 3.4 appear or not in the given image. AUCON is defined as follows,

$$AUCON = \frac{TP + TN}{\# \text{ of Evaluated AUs}} \quad (4.4)$$

where  $TP$  stands for true positives, meaning the number of AUs that both appear in the driving and generated image,  $TN$  stands for true negatives, namely the number of AUs that do not appear in the driving and generated image. The denominator in Equation 4.4 is the total number of evaluated facial action units, and it is set to 18.

### 4.3 Experimental Results and Analysis

Our experiments show that landmark coordinates of the driving image is a helpful heuristics for preserving the source’s identity and achieving accurate head poses. By directly using driving landmark coordinates to guide the alignment of individual landmarks in the generated image, our model achieved better performance on identity preserving and head pose accuracy on the VoxCeleb1 dataset, shown in Table 4.2.

Following evaluation protocols in [15, 16], we evaluate our baseline methods trained on VoxCeleb1 for reenacting unseen people from the CelebV test set. Shown in Table 4.3, our methods still have lower head pose error, however, the identity preserving capability is less ideal compared to Mesh Guided GCN [15]. The main reason is that the driving’s identity information was dismissed in the optical flow estimation stage of [15]. The landmark GAN and style transfer module we proposed are aiming at alleviating the identity preserving problem. These methods are designed to leverage training data to improve the image quality, hence their evaluation are shown in separate tables. In general, both landmark GAN and

style transfer can improve the model’s identity preserving capability. When combined together, our method achieves better identity preserving capability while maintaining a lower head pose error. Regarding the performance of federated model, we evaluate the model with a ResNet backbone for optical flow estimation. Compared to the same model trained by centralized learning, the federated model shows poorer performance on evaluated metrics.

### 4.3.1 Self-reenactment

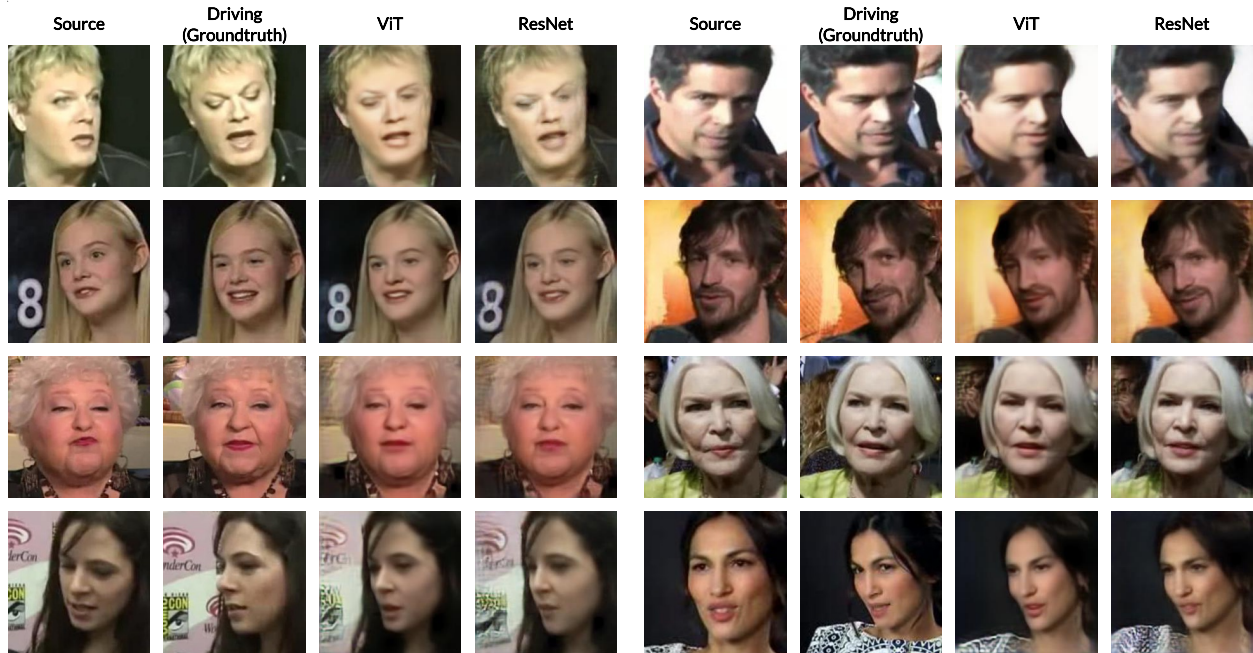


Figure 4.1: Self-reenactment on VoxCeleb1 dataset.

Table 4.2 shows models’ performance on the VoxCeleb1 dataset. Our method better preserves identities (higher CSIM) and shows lower error on head pose angles (lower PRMSE). This illustrates that coordinates of driving landmarks are a strong prior that can help models perform better on these two metrics. SSIM takes the structural similarities into consideration, which includes both the face and background of the image. Our method pays more attention on the face region, backgrounds in reenacted images are often distorted, resulting in a low score in SSIM. We believe the expression accuracy (AUCON) of our method is related to the presumption made in terms of reenacting individual facial landmarks. In the preprocessing stage, eyes and mouths for all people in the dataset are cropped into fixed sizes to ensure that the landmark reenactment model can handle varying landmark and camera movement in images. However, this also limits the model’s capability as there are

Table 4.2: Evaluation of Self-reenactment on VoxCeleb1

Model	CSIM $\uparrow$	SSIM $\uparrow$	PSNR $\uparrow$	PRMSE $\downarrow$	AUCON $\uparrow$
Mesh Guided GCN [15]	0.822	<u>0.739</u>	<b>30.394</b>	3.20	<b>0.887</b>
MarioNETte [16]	0.755	<b>0.744</b>	23.244	3.13	0.825
Monkey-Net [18]	0.697	0.734	23.472	3.46	0.770
FirstOrder [17]	0.813	0.723	<u>30.182</u>	3.79	<u>0.886</u>
NeuralHead-FF [23]	0.229	0.635	20.818	3.76	0.791
X2face [2]	0.689	0.719	22.537	3.26	0.813
ViT	<b>0.879</b>	0.608	29.297	<u>1.97</u>	0.767
ResNet	<u>0.878</u>	0.650	29.606	<b>1.58</b>	0.793

**Bold** shows the best results, second bests are underlined.  $\uparrow$  indicates the larger the value, the better the performance,  $\downarrow$  means otherwise.

Table 4.3: Evaluation of Reenacting Different Identities with Unseen Data on CelebV

Model	CSIM $\uparrow$	PRMSE $\downarrow$	AUCON $\uparrow$
MarioNETte [16]	0.520	3.41	<u>0.710</u>
Mesh Guided GCN [15]	<b>0.635</b>	3.41	0.709
Monkey-Net [18]	0.451	4.81	0.584
FirstOrder [17]	0.462	3.90	0.667
NeuralHead-FF [23]	0.108	3.30	<b>0.722</b>
X2face [2]	0.450	3.62	0.679
ViT	0.525	2.95	0.694
ResNet	0.515	<b>2.35</b>	0.708

cases where landmarks cannot fit in the cropped region. For instance, a wide open mouth or a close-up camera can lead to a larger mouth region, the model may still try to fit the entire mouth into region we cropped, resulting in less accurate expression reenactment. This phenomenon is also observed when reenacting different identities.

### 4.3.2 Reenacting Different Identities

Table 4.3 shows the overall performance on the CelebV dataset for models trained only on the VoxCeleb dataset. As mentioned above, Mesh Guided GCN [15] excludes the driving’s identity information when reconstructing 3D face models, the optical flows are then estimated based on these 3D models, leading to better identity preserving in generated images. With

Table 4.4: Evaluation of Reenacting Different Identities with Models Trained on CelebV

Model	CSIM $\uparrow$	PRMSE $\downarrow$	AUCON $\uparrow$
X2Face [2]	0.467	8.12	0.611
ViT	0.568	2.77	<u>0.692</u>
ViT+LGAN+Style	<u>0.653</u>	<u>2.66</u>	0.675
ResNet	0.570	<b>2.57</b>	<b>0.695</b>
ResNet+LGAN+Style	<b>0.661</b>	2.68	0.672

**Bold** shows the best results, second bests are underlined.  $\uparrow$  indicates the larger the value, the better the performance,  $\downarrow$  means otherwise.

the direct guidance of landmark locations, our method shows more accurate head poses. However, due to the fact that these landmark locations do not reflect the identities of source images, our method performs poorer than Mesh Guided GCN in terms of identity preserving.

The proposed landmark estimation and style transfer methods rely on learning from training samples to assist the image generation process, we then trained these models on the CelebV dataset. X2Face [2] is also trained from scratch on CelebV for comparison. Shown in Table 4.4, X2Face’s identity preserving capability is slightly improved compared to its performance in Table 4.3, however, the head pose error significantly increases. We find that X2Face has difficulty converging when trained on a smaller dataset such as CelebV. Our method achieves better identity preserving and lower head pose error with the help of the proposed landmark GAN and style transfer module.

### 4.3.3 ViT vs ResNet

In general, the ResNet variant of our method performs slightly better than its ViT counterpart. We believe this is because the optical flow estimated by either ResNet or ViT still operates on feature maps extracted by a CNN, ResNet is more compatible with this feature representation as it is also a CNN based model. However, Vision Transformer is still promising for face reenactment. When evaluated on ImageNet [35] with 10 million images for training, a Vision Transformer with 86 million parameters is outperformed by ResNet-50 with only 25 million parameters. In our case, the ViT head for optical flow estimation has 19 million parameters while the ResNet head has 21 million parameters. Our results show that the performance difference between these two models is negligible, a future study on Vision Transformer based image generator for face reenactment is worth investigating.

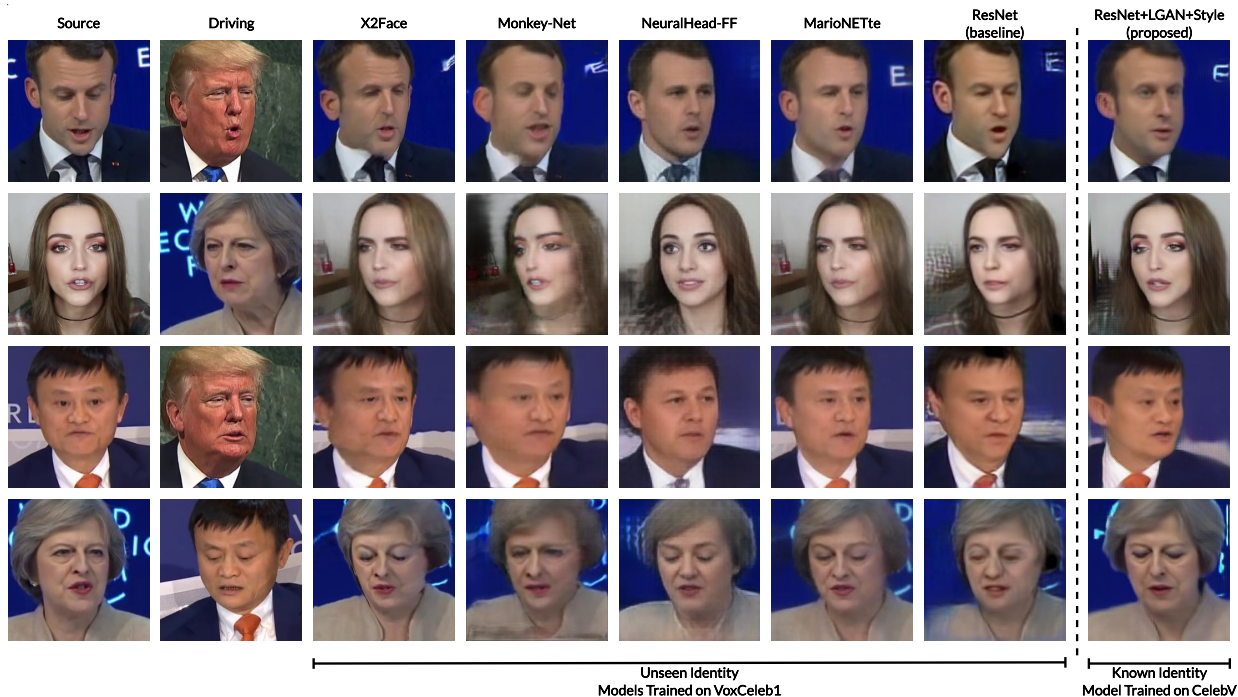


Figure 4.2: Comparison of Reenacting Different Identities on CelebV.

#### 4.3.4 Landmark Estimation and Style Transfer

Table 4.5 shows ablation study on proposed landmark estimation methods. Landmark Style Transfer, denoted by LSt, is a crude way of estimating landmark coordinates, it achieves the best identity preserving among our methods, but it also significantly hinders the pose and expression accuracy. Landmark Conditional GAN (LGAN), on the other hand, better balances these metrics.

Table 4.5: Evaluation of Landmark Estimation for Reenacting Different Identities on CelebV

Model	CSIM $\uparrow$	PRMSE $\downarrow$	AUCON $\uparrow$
ViT	0.568	2.77	<u>0.692</u>
ViT+LSt	<b>0.620</b>	3.87	0.646
ViT+LGAN	<u>0.619</u>	2.60	0.682
ResNet	0.570	<u>2.57</u>	<b>0.695</b>
ResNet+LSt	0.616	3.78	0.650
ResNet+LGAN	0.614	<b>2.49</b>	0.687

Table 4.6 shows the ablation study on style transfer. The model named "Style" is a baseline

Table 4.6: Evaluation of Style Transfer for Reenacting Different Identities on CelebV

Model	CSIM $\uparrow$	PRMSE $\downarrow$	AUCON $\uparrow$
Style	<b>0.647</b>	4.75	0.646
ViT	0.568	<u>2.77</u>	<u>0.692</u>
ViT+Style	0.587	3.22	0.668
ResNet	0.570	<b>2.57</b>	<b>0.695</b>
ResNet+Style	<u>0.606</u>	2.97	0.670

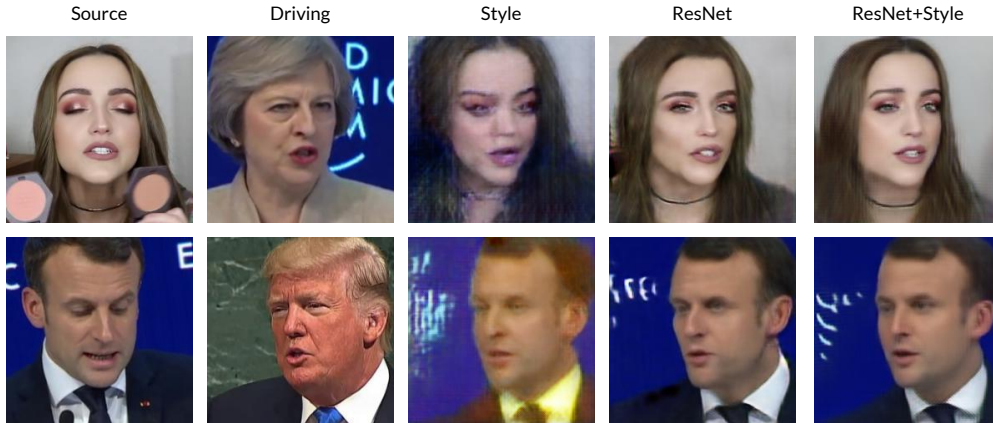


Figure 4.3: Optical flow combined with style transfer improved the quality of generated images.

model without optical flow estimation, it’s reenactment process solely relies on the image generator and style transfer branch in Figure 3.6. Although this model shows better identity preserving capability, it generates images with the poorest quality. Facial textures in generated images are often a mixture of the source and driving image. Although we explicitly exclude the RGB information from the style transfer input by using one-channel landmark heatmaps instead, the model ”memorizes” the connection between landmark heatmaps and their corresponding color images due to the self-supervised nature of the training stage. Evaluation metrics also show that style transfer has a similar effect on our method, namely promoting the identity preserving capability at the cost of head pose and expression accuracy. However, this does not reflect the real contribution of style transfer. As shown in Figure 4.3, faces generated by non-style-transfer methods are distorted because of the warp operation, style transfer can help our model revert unnecessary distortion on faces, generating more realistic images.

Table 4.7: Evaluation of Federated Face Reenactment on CelebV

Model	CSIM $\uparrow$	PRMSE $\downarrow$	AUCON $\uparrow$
ResNet+Style (C)	0.606	2.97	0.670
ResNet+Style (F)	0.555	3.47	0.665
ResNet+Style+LGAN (C)	<b>0.661</b>	<b>2.68</b>	<b>0.672</b>
ResNet+Style+LGAN (F)	0.601	3.57	0.672

(C) denotes centralized learning, (F) is federated learning.

### 4.3.5 Federated Learning vs Centralized Learning

Overall, models trained through centralized learning perform better than federated learning. Table 4.7 shows the evaluation results of federated learning on the CelebV dataset. We compared models trained through different learning paradigm, we also evaluated how the landmark conditional GAN performs in the federated setting. With the help of our landmark conditional GAN, federated model’s performance on identity preserving has been improved, however, unlike centralized models, the head pose error slightly increases.



Figure 4.4: Comparison between federated model and centralized model.

Figure 4.4 shows the comparison between federated and centralized models. Common problem in images generated by federated models are shape distortion, color inaccuracy and identity mismatch. These problems are reflected in Figure 4.4. The cause of these problems are rooted in the model aggregation of federated learning. The algorithm introduced in Chapter 3.6 aggregates client models by linearly combining model parameters. However, the parameter space of the optimal model can be highly non-linear, therefore aggregating client models through linear summation is equivalent to linearly approximating the optimal model, resulting in errors mentioned above.



## 5 Conclusion

We propose a novel face reenactment method guided by generative landmark coordinates. We evaluate our method in the following reenactment scenarios:

- **Self-reenactment.** In this scenario, the source and the driving image are taken from the same person, which allows us to directly use landmark coordinates in the driving image to guide reenactment as the driving image is also the groundtruth for the generated image. We evaluated our method on the VoxCeleb1 dataset and compared against existing methods which are also evaluated through the same protocol. We show that images generated by our method are more similar to the input image’s identity, and our method has lower head pose error compared to others.
- **Reenacting Different Identities.** In this scenario, the identities of the source and the driving image are different from each other. This scenario can be further divided into two settings: 1. reenacting different and unseen identities; 2. reenacting different but known identities. We follow the evaluation protocols of existing research and evaluated our method on the CelebV dataset.

For reenacting different and unseen identities, since we should not learn from the unseen data, we use our self-reenactment model trained on VoxCeleb1 for evaluation. Our method shows competitive performance on identity preserving and expression accuracy, and lower head pose error compared to existing methods, indicating that the heuristic of using driving landmark coordinates to guide face reenactment is beneficial for accurately reenacting head movement.

For reenacting different but known identities, we propose various modules to alleviate the identity preserving problem in face reenactment, including landmark style transfer, landmark conditional GAN, and style transfer in the image generator. We trained our models on the CelebV dataset and evaluated them on a test set similar to the one used for the above unseen scenario. By conducting ablation study on each module we proposed, we show that each of them is capable of enhancing generated images. When proposed modules are combined, we demonstrated that our models witnessed a significant increase in identity preserving while maintaining a lower head pose error.

- **Federated Face Reenactment.** We adapted the CelebV dataset for federated learning and applied the FedGAN algorithm for this task. We show that our model can still learn to perform face reenactment in the federated learning setting, however, the

quality of generated images are not on par with those generated by models trained through centralized learning.

## 5.1 Limitations

Given the experimental results in Chapter 5, we conclude major limitations of proposed method as follows:

- **Difficulty with Unseen Identities.** When evaluated on unseen data from the CelebV dataset, MeshGCN’s [15] performance on identity preserving significantly surpasses other methods including ours. Although we proposed landmark conditional GAN to alleviate this problem, it can only inference landmark coordinates for known identities. Which means that our landmark conditional GAN is inapplicable to the scenarios of reenacting unseen faces.
- **Difficulty with Different Videos of the Same Identity.** This problem is unique to the style transfer branch in the image generator. Given an identity included in the VoxCeleb1 dataset, more than one video clip may be sampled for this person. The person’s appearance is not necessarily the same in these videos due to changes in environment lighting, accessories, and facial hair. For instance, the same person may have beard in one video clip while being clean-shaven in another clip. The style transfer branch tends to blend these two facial hair styles and generate faces with slightly dark areas around the mouth regions. In order to improve the style transfer branch’s awareness of varying facial features, we tried to incorporate the identity embedding similar to our landmark GAN into the branch. However, unique appearance features found in the input are still missing in generated images.
- **Lack of Facial Action Unit Intensity.** The landmark conditional GAN only considers whether a facial action unit appears or not, it does not consider the intensity of such AU, resulting in less expression accuracy in generated images when the driving face has an intense expression.
- **No Constraints on Eye Gaze.** Our method does not consider the direction in which the eyes in the driving image gaze, resulting in the phenomenon that eyes in generated images may be looking at arbitrary directions. Accurately reenacting eye gaze directions is not only beneficial for improving the realism of generated images, but it is also crucial for real world application such of film making. Eye gaze constrains

are often neglected in image warping-based methods, however, the authors of Deep Video Portraits [11], a render-based method, did consider this problem and proposed a solution by specifically feeding the eye gaze direction into the image generator.

- **Sub-optimal Model Aggregation.** As mentioned in Chapter 5.3.5, our method aggregate client models by taking a linear combination of this, which may not reflect the optimal update direction of parameters. Therefore the performance of federated learning model is not as good as its centralized counterpart.

## 5.2 Future Work

Based on limitations discussed earlier, we argue that one possible future work is to extend proposed method to unseen faces. This may require models trained for more general tasks such as face recognition to provide useful features for unseen data. Specifically, our method requires the model to learn identity embeddings from training data, which may be replaced by a pretrained face recognition model which can generalize to arbitrary faces.

To tackle the problem with the style transfer branch, a more refined and subtler approach is needed. Our method modifies the entire feature maps in the image generator. In contrast, the authors of [53] investigated semantic regions in the GAN image generator and managed to control the appearance in generated image. An approach like this may not only be the solution to the problem with our style transfer branch, it could also achieve accurate eye gaze reenactment as this method can precisely modify the eye regions in generated images.

Another topic worth further investigation is improving the expression accuracy of generated images. As discussed earlier, our landmark conditional GAN only considers whether certain facial action units appeared or not in the driving image, but it does not take the intensity of corresponding facial action units into consideration. The outcome of this is that our model can not accurately reenact intense expressions such as one with a wide open mouth. Our model tends to generate an image with an moderately opened mouth instead. One solution to this could be that we only use landmark conditional GAN for alignment, but behaviors of individual landmarks in the driving image are retained, thus the expression intensity is intact.

Lastly, we need better aggregation strategy for federated face reenactment. Attention-based and boosting methods are all promising ways to solve the problem. It also worth mentioning that investigating Vision Transformer as an image generator for face reenactment is also a possible task. Currently we rely on CNN as the image generator because the warping

operation is essential and it is not defined on image representations in a Vision Transformer.

## References

- [1] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, “Face2face: Real-time face capture and reenactment of rgb videos,” in *CVPR*, 2016.
- [2] O. Wiles, A. S. Koepke, and A. Zisserman, “X2face: A network for controlling face generation using images, audio, and pose codes,” in *ECCV*, 2018.
- [3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *CVPR*, 2009.
- [4] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, “Ms-celeb-1m: A dataset and benchmark for large-scale face recognition,” in *ECCV*, 2016.
- [5] Q. Meng, F. Zhou, H. Ren, T. Feng, G. Liu, and Y. Lin, “Improving federated learning face recognition via privacy-agnostic clusters,” in *ICLR*, 2022.
- [6] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, “Voxceleb: Large-scale speaker verification in the wild,” in *Computer Science and Language*, 2019.
- [7] W. Wu, Y. Zhang, C. Li, C. Qian, and C. C. Loy, “Reenactgan: Learning to reenact faces via boundary transfer,” in *EVVC*, 2018.
- [8] Y. Nakashima, T. Yasui, L. Nguyen, and N. Babaguchi, “Speech-driven face reenactment for a video sequence,” in *IEEE Transactions on Media Technology and Applications*, 2020.
- [9] J. Thies, M. Elgharib, A. Tewari, C. Theobalt, and M. Nießner, “Neural voice puppetry: Audio-driven facial reenactment,” in *ECCV*, 2020.
- [10] Y.-T. Cheng, V. Tzeng, Y. Liang, C.-C. Wang, B.-Y. Chen, Y.-Y. Chuang, and M. Ouhyoung, “3d-model-based face replacement in video,” in *SIGGRAPH*, 2009.
- [11] H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, M. Nießner, P. Pérez, C. Richardt, M. Zollöfer, and C. Theobalt, “Deep video portraits,” *ACM Transactions on Graphics*, 2018.
- [12] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, “What makes tom hanks look like tom hanks,” in *ICCV*, 2015.

- [13] D. Vlastic, M. Brand, H. Pfister, and J. Popović, “Face transfer with multilinear models,” *ACM Trans. Graph.*, 2005.
- [14] V. Blanz and T. Vetter, “A morphable model for the synthesis of 3d faces,” in *SIGGRAPH*, 1999.
- [15] G. Yao, Y. Yuan, T. Shao, and K. Zhou, “Mesh guided one-shot face reenactment using graph convolutional networks,” in *28th ACM International Conference on Multimedia*, 2020.
- [16] S. Ha, M. Kersner, B. Kim, S. Seo, and D. Kim, “Marionette: Few-shot face reenactment preserving identity of unseen targets,” in *AAAI*, 2020.
- [17] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, “First order motion model for image animation,” in *Advances in Neural Information Processing Systems*, 2019.
- [18] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, “Animating arbitrary objects via deep motion transfer,” in *CVPR*, 2019.
- [19] 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 *arXiv: 2102.03984*, 2021.
- [20] X. Zeng, Y. Pan, M. Wang, J. Zhang, and Y. Liu, “Realistic face reenactment via self-supervised disentangling of identity and pose,” in *AAAI*, 2020.
- [21] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu, “Spatial transformer networks,” in *NIPS*, 2015.
- [22] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, “Generating 3D faces using convolutional mesh autoencoders,” in *ECCV*, 2018.
- [23] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, “Few-shot adversarial learning of realistic neural talking head models,” in *ICCV*, 2019.
- [24] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, “Openface 2.0: Facial behavior analysis toolkit,” in *13th IEEE International Conference on Automatic Face Gesture Recognition*, 2018.
- [25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *NIPS*, 2014.

- [26] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” in *arxiv.1411.1784*, 2014.
- [27] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” in *arxiv:1411.1784*, 2014.
- [28] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, “Generative adversarial text to image synthesis,” in *ICML*, 2016.
- [29] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *CVPR*, 2017.
- [30] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, “Video-to-video synthesis,” in *NeurIPS*, 2018.
- [31] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, “High-resolution image synthesis and semantic manipulation with conditional gans,” in *CVPR*, 2018.
- [32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *NIPS*, 2017.
- [33] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *arxiv.1810.04805*, 2018.
- [34] T. Brown and B. e. a. Mann, “Language models are few-shot learners,” in *Advances in Neural Information Processing Systems*, 2020.
- [35] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *arXiv:2010.11929*, 2020.
- [36] Y. Liu, Y. Zhang, Y. Wang, F. Hou, J. Yuan, J. Tian, Y. Zhang, Z. Shi, J. Fan, and Z. He, “A survey of visual transformers,” in *arXiv: 2111.06091*, 2021.
- [37] F. Zhu, Y. Zhu, L. Zhang, C. Wu, Y. Fu, and M. Li, “A unified efficient pyramid transformer for semantic segmentation,” in *ICCV Workshops*, 2021.
- [38] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *ICCV*, 2021.

- [39] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, “Going deeper with image transformers,” in *ICCV*, 2021.
- [40] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” *ACM Trans. Intell. Syst. Technol.*, 2019.
- [41] P. Kairouz and H. B. e. a. McMahan, “Advances and open problems in federated learning,” in *arxiv.1912.04977*, 2019.
- [42] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, “Communication-Efficient Learning of Deep Networks from Decentralized Data,” in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.
- [43] S. Ji, T. Saravirta, S. Pan, G. Long, and A. Walid, “Emerging trends in federated learning: From model fusion to federated x learning,” in *arxiv.2102.12920*, 2021.
- [44] M. Rasouli, T. Sun, and R. Rajagopal, “Fedgan: Federated generative adversarial networks for distributed data,” in *arxiv.2006.07228*, 2020.
- [45] X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in *ICCV*, 2017.
- [46] W. V. F. Ekman, P., “The facial action coding system: A technique for measurement of facial movement,” 1978.
- [47] A. Odena, V. Dumoulin, and C. Olah, “Deconvolution and checkerboard artifacts,” *Distill*, 2016.
- [48] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and improving the image quality of stylegan,” in *CVPR*, 2020.
- [49] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in *arXiv: 1511.06434*, 2016.
- [50] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in *ECCV*, 2016.
- [51] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” in *IEEE Transactions on Image Processing*, 2004.



- [52] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” in *CVPR*, 2019.
- [53] E. Collins, R. Bala, B. Price, and S. Ssstrunk, “Editing in style: Uncovering the local semantics of gans,” in *CVPR*, 2020.