Internship Thesis Report

Face Swapping and Deepfake Video

Author: Thu Hien LE

thuhienle2806@gmail.com

October 3, 2025

Keywords:
face swapping, diffusion models, image editing

Conducted under the supervision of:

Prof. **Christophe CHARRIER**
Prof. **Emmanuel GIGUET**
Prof. **Maxime BÉRUBÉ**

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisors, professor Christophe CHARRIER, professor Maxime BÉRUBÉ, and professor Emmanuel GIGUET, because of their invaluable guidance, patience, and profound knowledge throughout this research. They always encourage me to explore new ideas, even if they do not always come up with something satisfactory, they are crucial for my development as a researcher. I am also deeply grateful for the support from my supervisors and GREYC Laboratory for providing me the computational resource, without which this project would not have been possible.

I extend my gratitude to my program coordinators, professor Mihai MITREA and professor Sahar HOTEIT, for opening the first door for me to pursue my studies in France. I am also grateful for all the professors in the program of Multimedia Networking: Coding, Security, and Deep Learning; the knowledge they conveyed gave me the confidence to seek out new chances for myself.

I would like to express my deepest appreciation to my family in my homeland Vietnam for believing in me throughout my time studying abroad.

Finally, I would like to thank my friends, who have always been there to support me. In particular, I am deeply grateful to my dear friend, Nhat, for always standing by my side and sharing not only in my happiness and but also in moments of panic.

Abstract

Abstract: Face swapping, or deepfake generation, has achieved remarkable performance. However, a critical challenge persists in balancing the three main objectives: preserving the source identity, controlling the target's attributes, such as pose and expression, and generating a photorealistic image. Existing high-fidelity models, while powerful, still struggle with photometric inconsistencies like mismatched lighting and skin tones and can fail to perfectly preserve the target's precise attributes.

This thesis introduces a novel, three-stage pipeline designed to address these limitations by systematically disentangling and refining different aspects of the face swapping process. For **Lighting transfer module**, we normalize illumination by decomposing images to achieve a more accurate relighting method. For **Skin tone refinement**, we introduce a method to identify a semantic *skin direction* in the $\mathcal{W}_+$ latent space using a 3D Morphable Model (3DMM), ensuring precise skin tone transfer. Finally, for identity and attribute preservation, inspired by a powerful face swapping model, named REFace, we deeply investigate the significance of loss functions. We also incorporate a differentiable landmark loss function as a naive explicit control over pose and expression within our **REFace tuning module**.

Through extensive quantitative and qualitative results of our experiments, we demonstrate that our whole pipeline considerably outperforms the baseline in terms of attribute preservation and realistic results. Our pipeline achieves a state-of-the-art FID of **7.160** while also obtaining superior performance in $Pose$, $Expression$, and $SSIM$ metrics. These results confirm that our approach successfully generates highly realistic and coherent results that are robust to variations of lighting conditions and skin tone.

Keywords: *face swapping, diffusion models, image editing.*

Contents

List of Figures

List of Tables

1. Introduction

Contents

This chapter serves as the foundation for the thesis, establishing the context and importance of the research presented. We begin by providing a broad motivation for this work, exploring the diverse real-world applications of the face swapping problem. Subsequently, in section 1.2, we present the overall layout of the thesis to provide a clear road map for readers.

1.1. Motivation

In recent years, the field of generative models has seen exponential development, with face swapping, or deepfake generation, emerging as one of the most compelling recognized applications. The ability to seamlessly transfer a person's identity from the source image onto a target has captured innovation across a diverse array of domains. The practical and creative applications include **entertainment, social media, content creation, virtual and augmented reality, data augmentation, AI fairness, and data privacy**, e.t.c.

Despite these advancements and the impressive realism of the modern methods, achieving a perfect face swap remains a significant technical challenge. The core challenge lies in compromising identity preservation, attribute control, and photorealism. State-of-the-art models, while powerful, often struggle to obtain the perfect balance. They can exhibit perceptual artifacts that undermine the final result's credibility.

This thesis is motivated by the need to address two critical limitations observed in existing high-fidelity models:

- **Photometric inconsistency**: Many models fail to perfectly harmonize the swapped face with the target's environment, resulting in mismatched lighting conditions and skin tones.
- **Imperfect attribute preservation**: The accurate facial expression and head pose of the target face are not perfectly preserved, leading to un-photorealistic results.

Therefore, this research is driven by creating a pipeline that can systematically address these specific limitations. By separating the challenges of lighting and skin tone, our objective is to push the quality of realism and create a face swapping model that is not only coherent but also visually impressive.

1.2. Thesis Layout

The rest of this report is organized as follows:

- Chapter 2 explains technical background and reviews existing research that inspired this work.
- Chapter 3 describes the proposed methods, which constitutes our systematical pipeline for generating swapped images.
- Chapter 4 details the experiments conducted to confirm and evaluate the hypothesis presented in the previous chapter.
- Chapter 5 concludes the report by discussing advantages and limitations of the proposed methods, as well as outlining possible future improvements.

2. Related Works

Contents

This chapter reviews the technical background and related works which this thesis is built upon. We begin in section 2.1 by exploring the concept of image decomposition and its application to face relighting, a critical step for addressing the challenge of photometric consistency. Section 2.2 presents the powerful paradigm of GAN inversion-based image manipulation and explains how encoders like pSp and e4e project images into a latent space where certain attributes can be edited. This section will also cover the work of InterfaceGAN on learning disentangled representations, which forms basic of our skin tone refinement method. Finally, in section 2.3, we review the main face swapping backbone, REFace, by explaining the principles of Diffusion Models and detailing the specific architecture of this framework.

2.1. Face Relighting with Geometrically Consistent Shadow

Face relighting with geometrically consistent shadow [1] was introduced to address the face relighting problem. Based on the intrinsic decomposition technique shown in Fig. 2.1 [5], it renders new images from fundamental components, such as the geometry, albedo, and lighting condition, which are estimated from a deep learning neural network. The formula of a decomposition process is described in Eq. (2.1).

$$L_o(\mathrm{x},\omega_o) = a(\mathrm{x}).s(\mathrm{x}) \tag{2.1}$$

where $L_o(\mathrm{x},\omega_o)$ is the intensity of the pixel at the position (x), $a(\mathrm{x})$ and $s(\mathrm{x})$ are the albedo at that pixel and the shading, which represents the amount of light hitting the surface at that pixel, respectively.

Fig. 2.2 illustrates the workflow of relighting a give input image $\mathbf{I}_t$ with the target lighting direction ω_t. With replicated decoders and a multilayer perceptron (MLP) following the encoder,

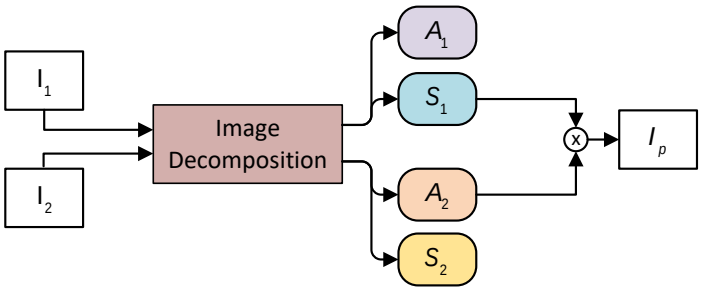


Figure 2.1: Image decomposition illustration.

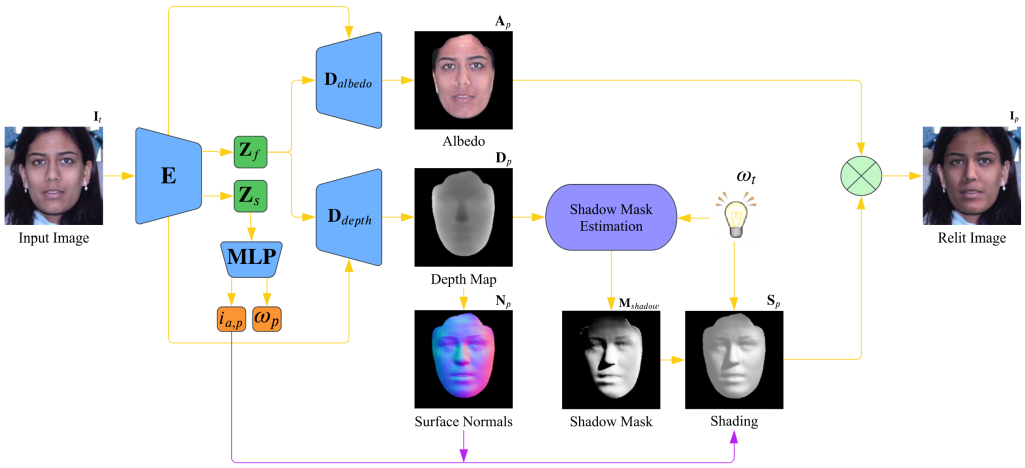


Figure 2.2: The Face relighting with geometrically consistent shadow framework in [1].

depth map $\mathbf{D}_p$, albedo $\mathbf{A}_p$, and lighting condition ω_p are estimated as principle intrinsic components. Thanks to the shadow mask estimation module, the authors adapt the principles of ray tracing [6] to generate the differentiable shadow mask $\mathbf{M}_{shadow}$ from $\mathbf{D}_p$ and ω_t. To be more specific, with each point on the face, a shadow ray is cast toward the light source. To determine if this ray is occluded by the facial geometry, they sample points along the ray and compute the minimum distance to the ray d_{min} using a cross product operator. The small value of d_{min} indicates an intersection, meaning that the point is in the shadow. This distance is mapped to the shadows between 0 and 1 using a sigmoid function to ensure the shadow estimation process is differentiable. The resulting shadows are consistent with the 3D shape of the face because they are derived directly from its geometry.

2.2. GAN Inversion-based Image manipulation

The target of image manipulation or image editing is to transform the content of a given image to achieve desired alterations or enhancements. With face image problems, this technique has significantly advanced in numerous applications such as data augmentation or entertainment,

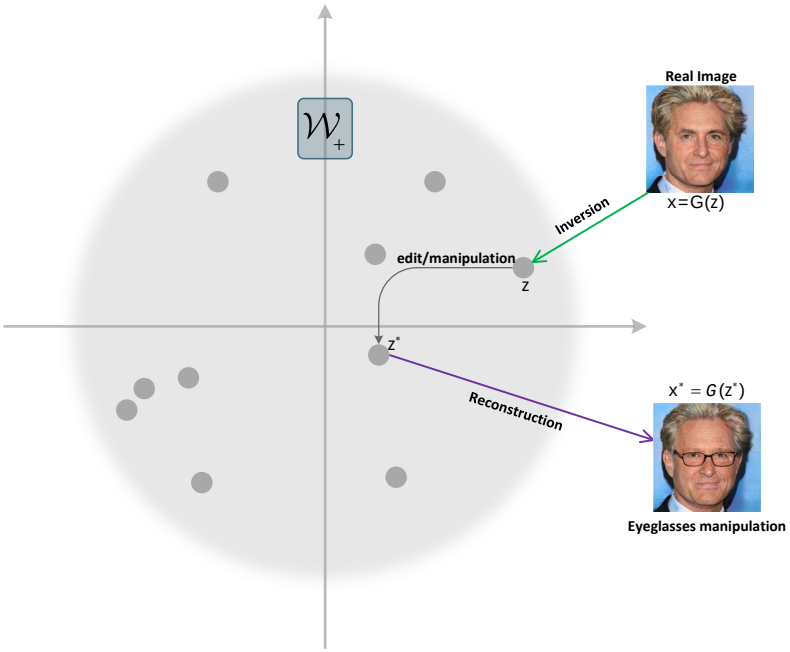


Figure 2.3: Illustration of GAN Inversion.

e.t.c. Recently, with the development of GAN family models, many works have tried to understand and control the latent space [7, 8], these methods follow the process of *invert first, edit later* [2]. As shown in Fig. 2.3, after an input image is inverted in the **latent space**, we can vary its latent code along a specific direction to semantically manipulate the corresponding attribute before feeding it into a pretrained StyleGAN generator to return the synthesized image.

This idea requires the quality of choice in designing which space is used to embed the image in. Numerous GAN Inversion methods have been developed using various latent spaces. In addition to the $\mathcal{Z}$ for general GAN, they introduce other different latent spaces for StyleGAN, called $\mathcal{W}$ and $\mathcal{W}_+$. A good latent space should be disentangled and easy to embed images in.

$\mathcal{Z}$ space : The generator of GAN models learns to map sampled points in the normal distribution ($z \sim \mathcal{N}(0, 1)$) to the image space. This latent space is applicable and used popularly in various research like DCGAN [9], PCGAN [10], e.t.c. However, this latent space is entangled; it means that changing one value of a latent code might affect multiple other attributes in a non-intuitive way. For example, only one altered value might change both the skin color and the background of the image.

$\mathcal{W}$ space : To eliminate the limitation of $\mathcal{Z}$ space, [8] introduces $\mathcal{W}$ as a space having a higher degree of freedom. By using a non-linear mapping network implemented with an 8 layer-perceptron, they map latent code in $\mathcal{Z}$ to $\mathcal{W}$. Thanks to this mapping network, the new latent is more disentangled and easier for downstream tasks [11].

$\mathcal{W}_+$ space : It is an extension of the $\mathcal{W}$ space. Rather than utilizing one single vector (size 1×512) to control all style input of the generator, this space allows concatenating 18 different layers (the latent code now has size 18×512) to be used in each layer of StyleGAN. This advantage results in a higher degree of freedom, more disentangled and highest expressiveness.

2.2.1. Encoders for image manipulation: pSp and e4e

Apart from choosing an efficient latent space to embed images in, we need a strong encoder that has the ability to match an input image to an accurate latent code in the latent space domain. Richardson et al. [2] and Tov et al. [12] introduced encoders of **pSp** and **e4e**, respectively.

To be more specific, the core problem is the reconstruction-editability trade-off in StyleGAN Inversion. While using a latent code (1×512) in the $\mathcal{W}$ space returns highly editable results, it is not efficient to reconstruct the original image accurately. On the contrary, using more disentangled latent code in $\mathcal{W}_+$ (18×512) yields satisfactory reconstruction results but sacrifices editability.

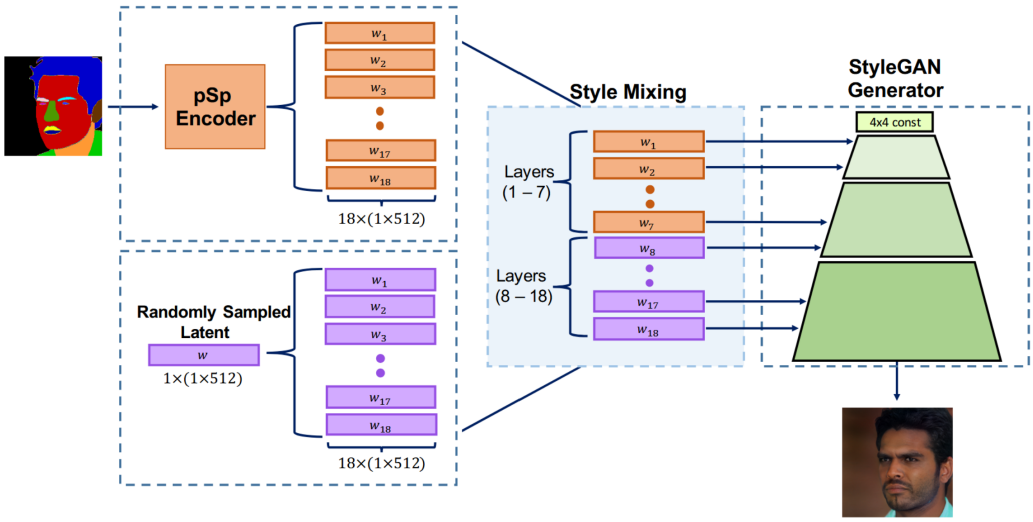


Figure 2.4: Style mixing for image generation with segmentation map input in [2]

In [2], the pSp introduces an encoder which is trained with a feature pyramid [13] to generate features at three different levels from which the styles are extracted by a **map2style** network. The final image is produced by feeding these hierarchically aligned styles into the generator. Overall in the translation process, input pixels are transformed into an intermediate style representation before converting to output pixels. Additionally, a technique named **style mixing** is designed to synthesize images. Given an input (such as a sketch or segmentation map), it is fed into the pSp encoder to achieve a set of 18 style vectors in $\mathcal{W}_+$ corresponding to different layers within the StyleGAN generator. Specifically, as can be seen in Fig. 2.4, the layers 1-7 are responsible for coarse level attributes, such as structure and pose of the face. Meanwhile, the layers 8-18 control fine level attributes, including texture, color, and lighting. Following this idea, the structural information from early layers of the latent code can be mixed with textural

information from a random style. This mixed vector is then the input of the StyleGAN generator to synthesize the final image, which effectively blends the structure of the input map with a random texture.

e4e [12] is introduced as a direct successor that builds upon the architecture of pSp. While psp prioritizes the reconstruction performance to reconstruct excellent real images, e4e's authors convince that inverting images away from the original $\mathcal{W}$ space results in lower perceptual quality as they are less editable. In other words, editability and perceptual quality are best achieved when an image is inverted to $\mathcal{W}$ with low variance between the different style vectors and each style vector should be within the distribution $\mathcal{W}$. To do that, it learns to make output latent codes very similar to the ones that StyleGAN was originally trained on. To be more specific:

- Firstly, rather than predicting 18 different style vectors like pSp, e4e predicts a single base style code and a series of small **offsets** to keep 18 style vectors very similar to each other and then the output latent code in a more stable and editable region.
- Secondly, instead of learning all 18 offsets at once, it is trained with a progressive scheme. The model learns offsets for the first few layers corresponding to the coarse details and then learns offsets for the middle layers and finally for the top layers to control medium and fine details.
- Finally, to make sure the code is in an efficient region, e4e uses a discriminator during training to classify *real* latent codes (from StyleGAN's mapping network) and *fake* ones generated by the e4e encoder.

2.2.2. Disentangled face representation learned by GAN: InterfaceGAN

The main reason we employ GAN Inversion is that it allows us to understand and manipulate certain attributes of images by editing the latent code in the latent space [11, 14, 15, 16]. In [11], they proposed a method called **InterfaceGAN**, which supports a way to edit facial attributes in images. Specifically, with a single attribute, they can vary a latent code z by moving it along the desired directions $\mathbf{n}$ with an intensity control coefficient α to obtain a new latent code:

$$z_{edit} = z + \alpha \times \mathbf{n} \tag{2.2}$$

This direction $\mathbf{n}$ is identified as the normal vector of the attribute's separating hyperplane. The hyperplane is defined as the boundary of the attribute and can be determined by off-the-shelf classification models. The process of identifying this boundary relies on a pretrained external **attribute score predictor** network, which assigns a quantitative score s to an image based on the intensity of the considered attribute. They created a dataset of (*latent vector*, *score*) pairs and then trained a linear Support Vector Machine (SVM) [17] on such data to find the hyperplane and its normal vector $\mathbf{n}$.

This research provides a powerful and intuitive framework for **disentangled** control, demonstrating that meaningful semantic directions can be discovered and utilized for precise facial

editing.

This approach is applicable with attributes having binary classes and already labeled in datasets, for instance, *smile* vs *no smile* or *male* vs *female*. The reason is that with these attributes, it is effective to use SVM to identify the hyperplane in the latent space such that all samples from the same side are with the same attribute. Additionally, determining the hyperplane of binary labeled attributes is plausible, as it might be easy to employ off-the-shelf classifiers as attribute score predictors. For instance, authors in [11] trained five independent linear SVMs on pose, smile, age, gender, and eyeglasses.

2.3. REFace: An unified framework for Face Swapping

2.3.1. Diffusion Models

Diffusion models are also designed based on the idea of manipulating the latent code in the latent space. They are trained to understand distribution of the data within this compressed space. Moreover, these models often use a hierarchical structure scheme, which allows them to learn both local representation and global features of the data.

Diffusion models are a relatively new but important technology in the field of face swapping or deepfake generation [18, 19, 20] and yield better performance than GAN-based methods because of the more stable and reliable training process. GANs frequently encounter "collapse mode" problems [20], where they fail to generate diverse results. In contrast, diffusion models have clearly defined learning objectives, which results in a smoother training experience.

A common type is the **denoising diffusion model** or vanilla diffusion model. This process begins by gradually adding random noise to an image sampled in the data distribution over several steps, like a Markov chain, until it is completely distorted. The main objective for the model is then to learn how to reverse this process and effectively remove the noise to restore the original image. This learning goal is represented by the following loss function:

$$\mathcal{L}_{denoise} = \mathbb{E}_{\mathrm{x}\sim p_{data}}\left[\frac{1}{2\sigma^2}\|\mathrm{x}-\tilde{\mathrm{x}}\|_2^2 + \frac{1}{2}\log\left(2\pi\sigma^2\right)\right] \tag{2.3}$$

where x is the ground truth, $\tilde{\mathrm{x}}$ is the denoised sample generated by the diffusion model at a time step t. σ^2 is the variance of the noise added to the input during the forward process. Euclidean distance is utilized here to measure the difference between the original image and the reconstructed version. Once the training process is completed, the model will be able to understand the latent space used for facial manipulation.

However, the vanilla diffusion models have to deal with heavy computational cost in both training and inference processes, since the diffusion process is directly operated on the pixel level ($\mathrm{x} \in \mathbb{R}^{3\times H\times W}$). To overcome this issue, **Latent Diffusion Models** (LDM) are introduced in [21] to compress the images into a latent space first, as demonstrated in Fig. 2.5, and the diffusion model is now trained on that latent space instead of image space. The diffusion loss

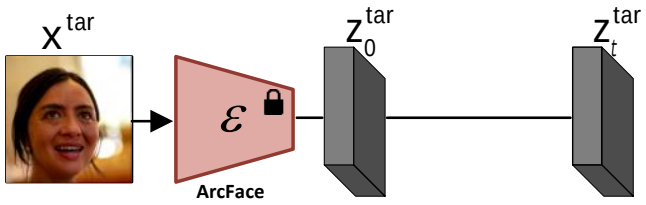


Figure 2.5: Illustration of latent diffusion models.

function becomes:

$$\mathcal{L}_{denoise} = \mathbb{E}_{z_0,\epsilon,t}\|\epsilon_\theta(z_t, t) - \epsilon\|_2^2 \tag{2.4}$$

where ϵ_θ is trained latent diffusion model, which predicts the noise at time step t.

Another widely used variant of LDM is conditional LDM, which is trained to generate images with the guidance of a side information. The conditional training loss is described in Eq. (2.5).

$$\mathcal{L}_{denoise\text{-}conditional} = \mathbb{E}_{z_0,c,\epsilon,t}\|\epsilon_\theta(z_t, c, t) - \epsilon\|_2^2 \tag{2.5}$$

where the condition vector c is usually an output of a pre-trained condition encoder, acting as a source of extra prior.

$$c = \mathcal{E}_c(\mathrm{x}) \tag{2.6}$$

2.3.2. REFace Architecture

The core of the REFace model is a conditional inpainting diffusion model. The objective of the model is to learn how to reconstruct a facial region within a mask realistically, guided by a rich set of conditioning features. This ensures obtaining the goal of face swapping, which not only looks real but also preserves the desired source identity. The training set can be broken down into three fundamental aspects: Input preparation, Condition generation and Training objectives.

Input preparation As illustrated in Fig. 2.6, the input for training REFace diffusion model consists of a target latent code (z_t^{tar}), an "inpainting image" (z^{inp}) generated by taking x_t^{tar} and applying a mask (m^{tar}) that has been slightly transformed using **Face Shape Augmentation** ($\mathcal{FA}$) with the following equation:

$$z^{inp} = \mathcal{E}(x^{tar} \otimes \mathcal{FA}(1 - m^{tar})) \tag{2.7}$$

Where $\mathcal{E}$ is a latent space encoder. The reason for employing this augmentation is to prevent the model from learning a "copy-paste" solution and to ensure it can handle variations. By changing the shape of the masked region, module $\mathcal{FA}$ forces the model to learn how to reconstruct facial features instead of memorizing how to fill a static shape. This results in a more robust face swapping capability.

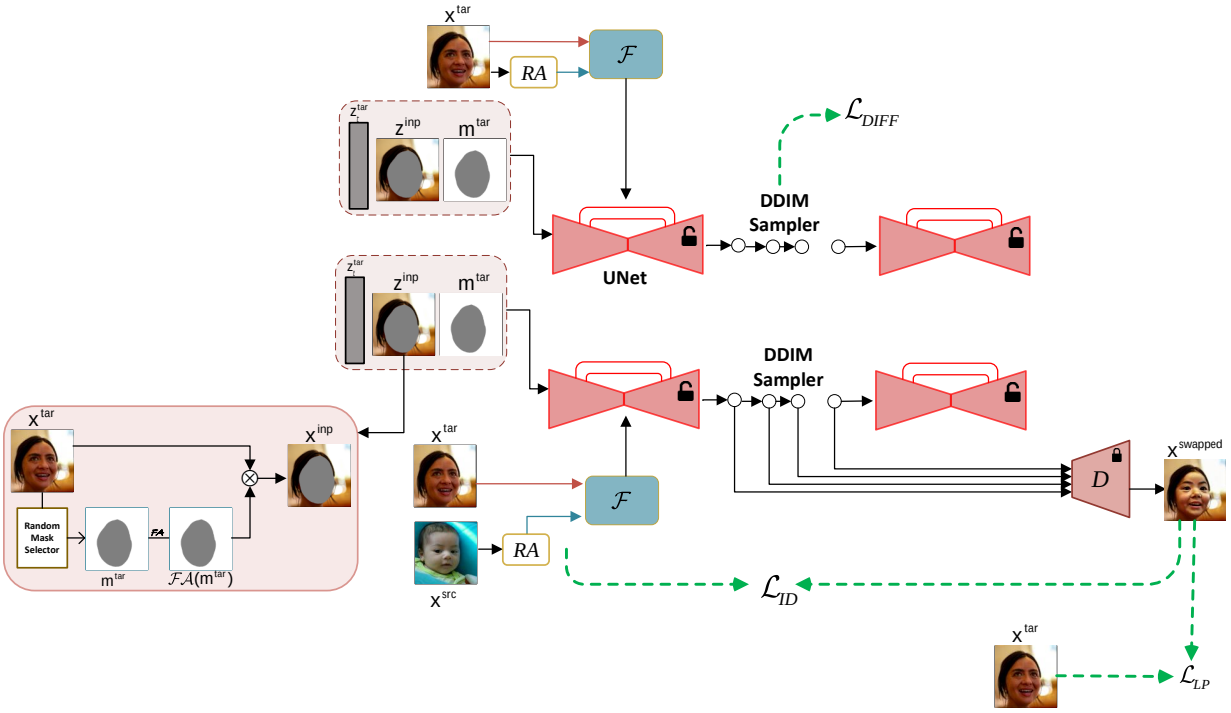


Figure 2.6: Training pipeline of the REFace architecture.

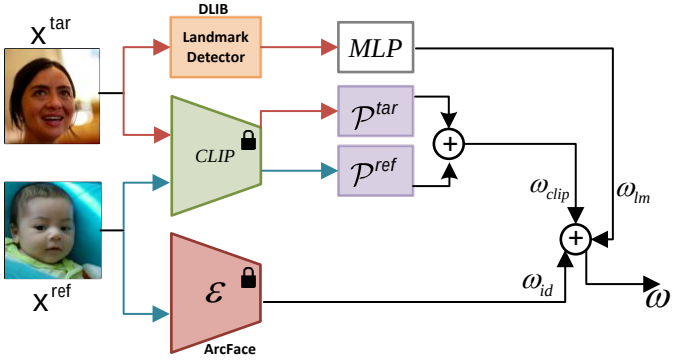
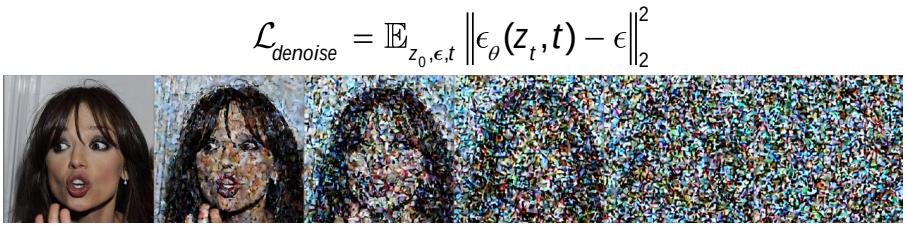
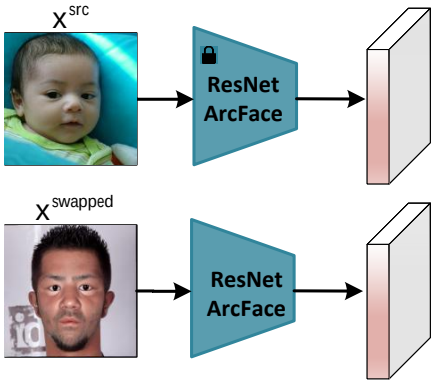


Figure 2.7: Condition generation module ($\mathcal{F}$) of the REFace architecture.

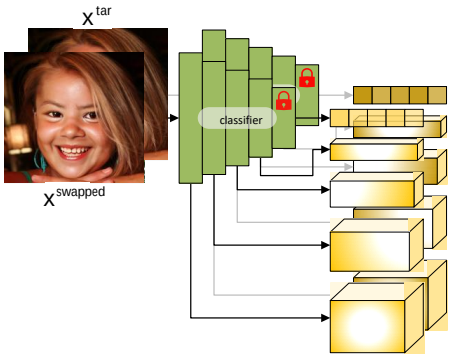
Condition Generation The instruction for the inpainting process comes from a conditional feature vector, which is denoted as the $\mathcal{F}$ module in Fig. 2.6. The process of producing the condition vector is illustrated in Fig. 2.7, to design a meaningful condition, authors in [22] not only use ID source features extracted from ArcFace [23] and target landmark features but also leverage the powerful **CLIP image encoder** [19] to extract richer information. To be more specific, they trained two linear projectors: a **reference projector** ($\mathcal{P}^{ref}$) to extract identity information from the reference image and a **target projector** ($\mathcal{P}^{tar}$) to extract target



(a) Diffusion loss at multiple sampling steps



(b) Cosine similarity of facial features



(c) VGG compares multi-level features in LPIPS loss

Figure 2.8: Training objectives of the REFace.

attributes such as pose and expression.

The final condition feature f is produced as four steps:

- **Step 1**: From the reference image x^{ref}, they extracted the CLIP feature (f^{ref}_{clip}), and the ArcFace identity feature (f_{id}).
- **Step 2**: From the target image x^{tar}, they extracted the CLIP feature (f^{tar}_{clip}), and the DLIB [24] facial landmark feature (f_{lm}).
- **Step 3**: The CLIP feature is formed by projecting the reference and target CLIP features by the following equation:

$$f_{clip} = \mathcal{P}^{ref} \circ f^{ref}_{clip} + \mathcal{P}^{tar} \circ f^{tar}_{clip} \tag{2.8}$$

- **Step 4**: Finally, condition f is a weighted average of all components:

$$f = \omega_{clip} \times f_{clip} + \omega_{id} \times f_{id} + \omega_{lm} \times f_{lm} \tag{2.9}$$

where, ω_{clip}, ω_{id}, and ω_{lm} are the weights to average the features. Additionally, the conditional feature is utilized as the key in each cross attention in diffusion U-Net [22].

Training objectives Fig. 2.8 demonstrates the loss functions utilized in training the REFace. To ensure high-fidelity outputs, they used the combination of several loss functions:

- **Diffusion loss function** ($\mathcal{L}_{Diff}$): The primary loss function of diffusion models, which is presented in 2.3.1. It typically is an ℓ_2 loss between the noise predicted by U-Net and the actual noise which was added in the diffusion process. This loss function forces the model to learn the denoising process.
- **Identity loss function** ($\mathcal{L}_{ID}$): To eliminate the lack of ID feature transferability and improve the ID preservation, this loss function is calculated as cosine similarity between facial features of source images and swapped results, which are extracted by ArcFace [23].
- **Perceptual loss function** ($\mathcal{L}_{LP}$): The LPIPS loss is computed between target images and the swapped ones. This encourages other attributes like background, lighting condition, expression, and head pose to be kept unchanged and visually similar to the original target. To be more specific, they employed the VGG backbone [25] to extract images and compare features at multiple levels to force them to be similar to each other.

For the whole training procedure, the total loss function is a weighted average of all loss function components:

$$\mathcal{L}_{total} = \mathcal{L}_{Diff} + \omega'_{ID} \times \mathcal{L}_{ID} + \omega'_{LP} \times \mathcal{L}_{LP} \tag{2.10}$$

where, ω'_{ID} and ω'_{LP} are the weights to average the loss functions. In the original paper, these values are set to 0.3 and 0.1, respectively.

3. Methodology

Contents

This chapter provides the core contributions of this thesis. We begin in section 3.1 by formally defining the problem, establishing the complex and objectives of the task that requires balancing identity preservation, attribute control, and realism. Following this, section 3.2 introduces our contributions: a novel, three-stage pipeline as the systematically solution for this problem. We will explain each module details. Firstly, we describe the Lighting transfer module, which is designed to solve photometric inconsistency by normalizing lighting conditions. Secondly, we present the Skin refinement module as the solution of accurate skin tone transfer while ensuring identity preservation. Finally, we detail the REFace tuning module, the final stage refining the output.

3.1. Problem Statement

Let I_{src} be the source image containing the desired identity and I_{tar} be the target image providing background, lighting conditions, and attributes such as head pose and facial expression. Let G_{swap} represent the face swapping generator. The objective is to generate a swapped image $I_{swapped}$ such that:

$$I_{swapped} = G_{swap}(I_{src}, I_{tar}) \tag{3.1}$$

The main challenges in this study is to compromise between three factors:

- Identity preservation: $I_{swapped}$ must faithfully represent the unique facial characteristic of the person in I_{src}.
- Attribute control: The non-identity attributes of the target images I_{tar} should be kept in the swapped results. This includes facial attributes, lighting conditions and background elements.
- Realism: The final image $I_{swapped}$ should be photorealistic, coherent, and free of visual artifacts. It should blend seamlessly with the background and be indistinguishable from the real image.

3.2. Proposal Pipeline

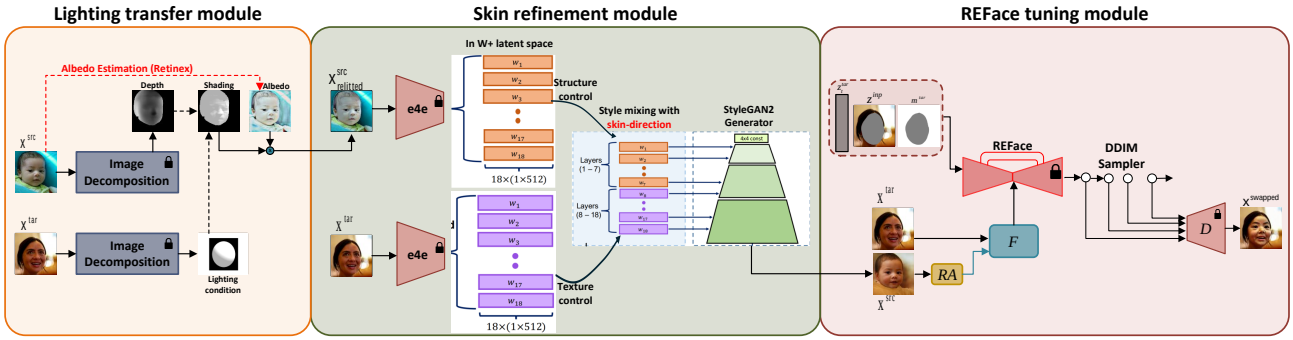


Figure 3.1: The architecture of the proposed face swapping pipeline.

As illustrated in Fig. 3.1, to address the challenges of identity preservation, attribute control, and realism in face swapping, we propose a novel three-stage pipeline. This approach systematically handles different aspects of the synthesis process: illumination transferring, blending identity and skin attributes in the latent space, and finally refining the output to achieve high-fidelity results.

In stage 1, our goal is to mitigate lighting inconsistencies between source images (I_{src}) and target ones (I_{tar}), which is a common cause of unrealistic blending. This process ensures that the source image is adapted to the target's lighting condition before the main synthesis occurs.

In stage 2, to perform the identity transfer while ensuring the skin tone and texture of the synthesized face match the target. We employ a skin refinement module to manipulate features in the disentangled $\mathcal{W}_+$ latent space of a pretrained StyleGAN-v2 generator. Additionally, we propose a methodology which involves identifying and transferring the specific **skin direction** in the latent space to ensure an accurate and natural skin tone blend.

Finally, we investigate the significance of loss functions to the performance of the REFace model. To do that, we conduct experiments with different weights of loss functions. Moreover, we hypothesize the impact of the landmark loss function in the stage of the REFace tuning module.

3.2.1. Lighting transfer module

As can be seen in Fig. 3.2, in this project, we adopted the architecture in [1] and applied its pretrained model to extract the lighting condition ω_t from the target image while simultaneously estimating albedo $\mathbf{A}_p$ and depth map $\mathbf{D}_p$ from the source image. The shading is then calculated from the source's geometry $\mathbf{D}_p$ under the target's lighting condition ω_t. Finally, the relit image is synthesized by combining this shading with the source's albedo using Eq. (2.1).

However, our qualitative analysis shown in Fig. 3.3 reveals a considerable limitation of this approach. The generated albedo $\mathbf{A}_p$ is not satisfactory; they retain a significant amount of the original lighting from the source image, which is also called *lighting leakage*.

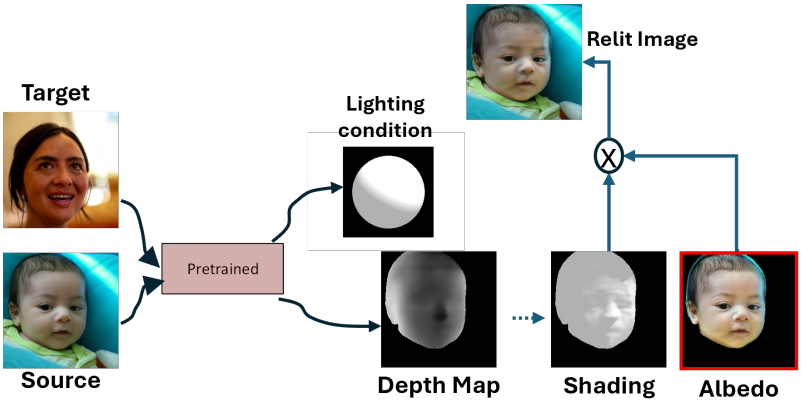


Figure 3.2: Diagram of the image relighting methodology.

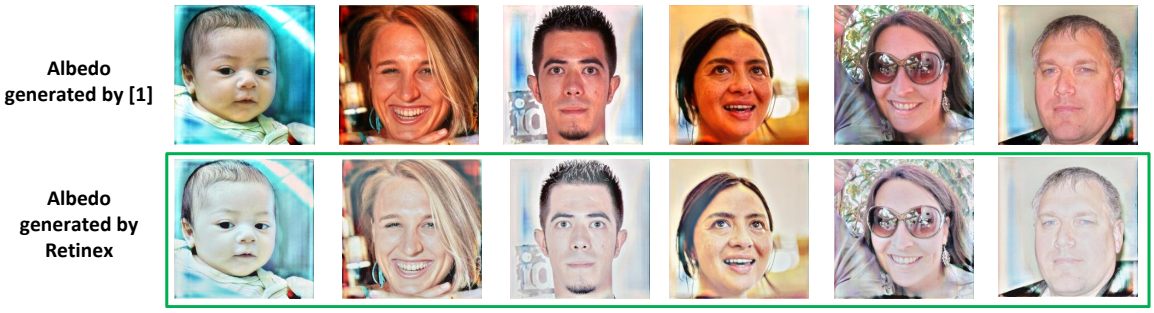


Figure 3.3: A qualitative comparison showing lighting leakage in the albedo maps from [1] (top row) and the cleaner, more accurate estimations from the proposed the Retinex-based approach (bottom row).

To address this issue, based on the idea of identifying and employing a more accurate albedo estimation method to ensure the quality of the image decomposing process, we propose to replace [1] by **Retinex theory**[26] in the albedo estimation step. The Retinex algorithm is designed to separate an image into its reflectance (albedo) and illumination components. We observe that Retinex-based approach will achieve a cleaner albedo that is a more faithful representation of the intrinsic properties, the results can be seen in the bottom row of Fig. 3.3.

We also conduct an experiment to evaluate the impact of the relighting process. To be more specific, for a given source and target pair, we consider two scenarios of swapping without(**w/o**) and with the lighting transfer module. We calculate a **relighting error heatmap** for each output. This heatmap represents the absolute difference in pixel values between the generated face and the target face. The lower error indicated by darker regions on the maps signifies that the lighting condition of the swapped face is more consistent with the illuminance of the target face.

As demonstrated in Fig. 3.4, across all test cases, the heatmaps for results without the lighting transfer module indicates a considerable mismatch between the lighting on the swapped result and the lighting on the target image. In contrast, the heatmaps of results generated with the lighting transfer module are consistently darker.

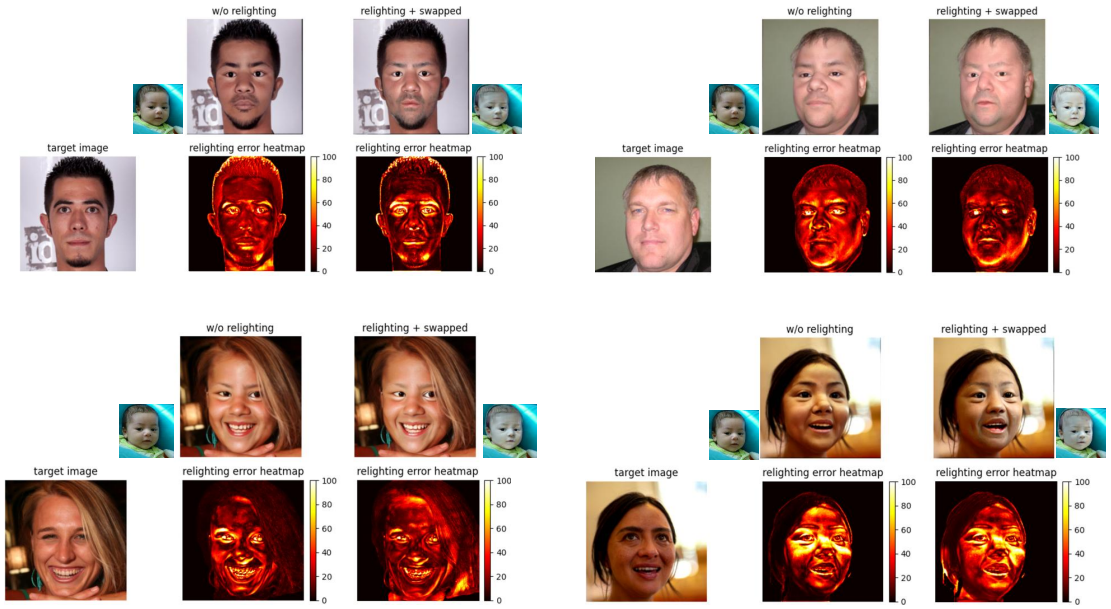


Figure 3.4: Quantitative result comparison of relighting performance. For each pair, the the result without relighting module (top left) shows a high error heatmap (bottom left), while our proposed method (top right) shows a significantly lower error in its corresponding heatmap (bottom right).

3.2.2. Skin refinement module

As illustrated in Fig. 3.5, in this project, our skin refinement module utilizes the e4e encoder, building upon the pSp architecture. Our objective is to transfer the skin tone from a target image x^{tar} to a source image x^{src} while keeping unique identity of the source. To achieve this, we employ a pretrained e4e model to embed both the source and target images into their respective latent codes within $\mathcal{W}_+$ space. After that, we perform style mixing presented in section 2.2.1, the first layers $(1-7)$ of the source latent code, which are responsible for structure identity, are combined with the later layers $(8-18)$ of the target latent code, as these layers control fine detail attributes including texture and color. This mixed latent vector is then fed into a pretrained StyleGAN2 generator to produce the skin-refined output image $x^{src'}$.

Simultaneously, to identify and transfer skin tone more accurately, the attribute direction-based image manipulation presented in section 2.2.2 is utilized to refine the result. However, our problem here is that the skin color is continuous and not considered as a binary attribute. Moreover, labels or scores of this attribute in public datasets are not available. This issue motivates us to propose a method to manipulate un-predefined attribute of skin color, allowing for controlled and continuous manipulation.

Because labels for skin tone are not available in practice, to identify the direction for this attribute without explicit annotations, we first need a quantitative proxy score. We propose using a pretrained **3D Morphable Model** (3DMM) introduced in [3] for this purpose. Specifically,

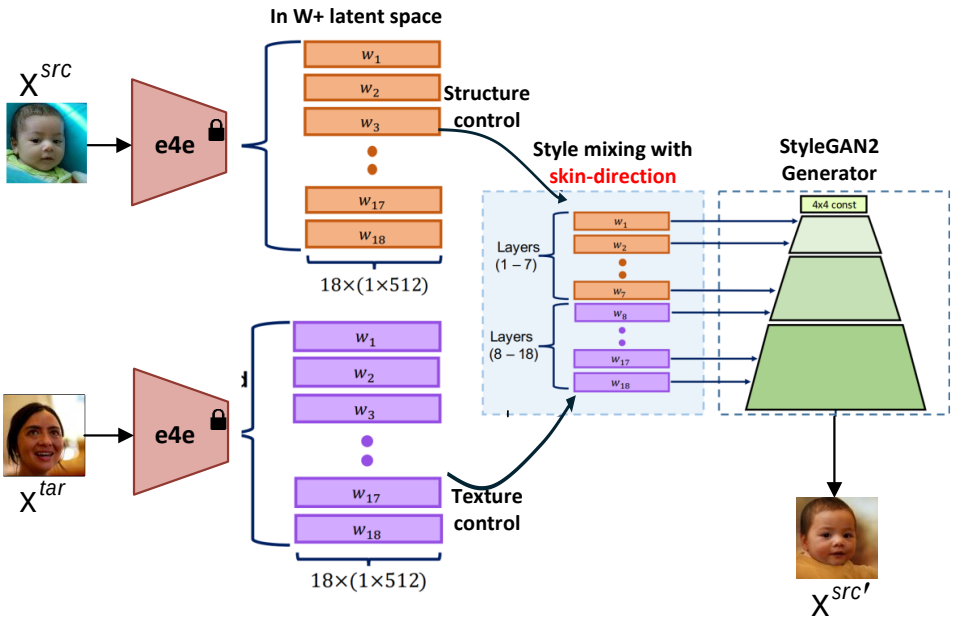


Figure 3.5: Flow of the skin refinement methodology.

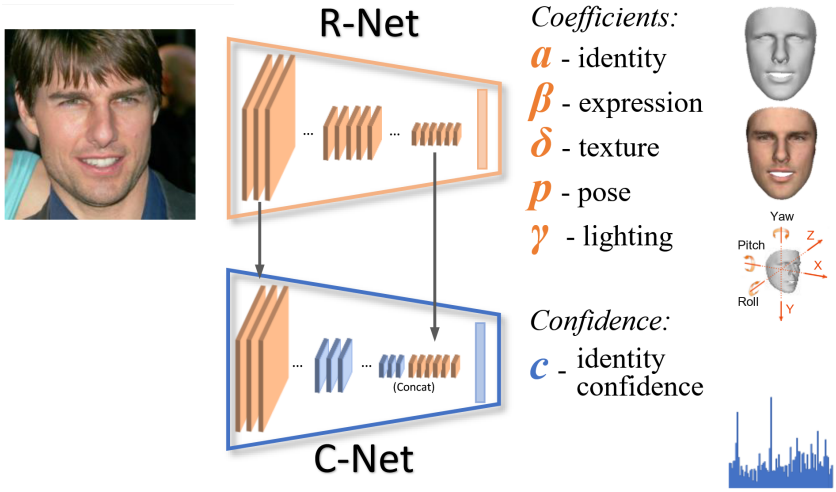
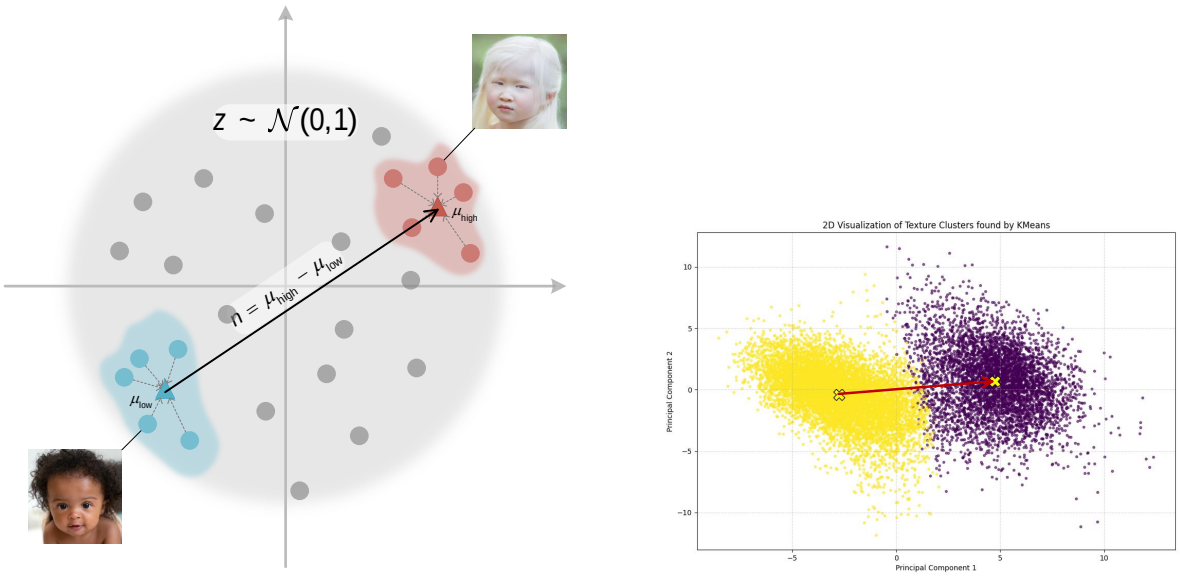


Figure 3.6: Texture coefficients are extracted by the 3DMM model. Figure is in [3].

we utilized a pretrained StyleGAN2 to generate a dataset containing 20,000 pairs of image and latent code. After that, we employ the 3DMM to extract parameters from the huge dataset. Among attributes, only texture coefficients, denoted by δ is considered, all other attribute coefficients are disregarded (Fig. 3.6). This skin coefficient is an 80-dimensional tensor ($\delta \in \mathbb{R}^{80}$). We use this tensor as high dimensional proxy score for the skin tone attribute, allowing us to sort and group images along this feature axis.

Following the principle of InterfaceGAN [11], the final step is to find a hyperplane in the



(a) (b)

Figure 3.7: Idea of identifying the skin direction in our project (a) and The process of determining the skin direction from 3DMM texture coefficients using K-Means (b).

StyleGAN $\mathcal{W}_+$ latent space that acts as a semantic boundary for the skin tone attribute and from that the direction can be determined as the hyperplane's normal vector. However, as we explain above, skin tone is continuous; training a linear SVM to find a separating hyperplane between two groups is infeasible. Alternatively, we propose a method to directly compute the skin tone direction by calculating the centroids of the two clusters separated by **K-Means** [27] algorithm; the idea can be seen in Fig. 3.7a.

As illustrated in Fig. 3.7b, two clusters naturally separate faces into two groups (*i.e.,* lighter and darker tones). The vector connecting the centroids of these clusters provides a robust direction for the skin direction. This identified vector **n** obtained by our method allows us to perform more precise edits in any give face by moving its latent code z along this direction with eq. (2.2).

To validate the effectiveness of our proposed method for skin tone manipulation, we conducted a series of qualitative experiments. Our objective is to perceptually evaluate whether incorporating the skin direction vector leads to a more accurate transfer of skin tone from the target image onto the source identity.

The qualitative results presented in multiple examples in Fig. 3.8 provide compelling visual evidence for the improvement of our proposed method. It is clear to be seen that our method consistently produces results where the skin tone more accurately matches the target images.

Without skin direction, the outputs appear to be an intermediate blend of the source and target skin tones. The influence of the source's skin color is often still visible, leading to an inaccurate

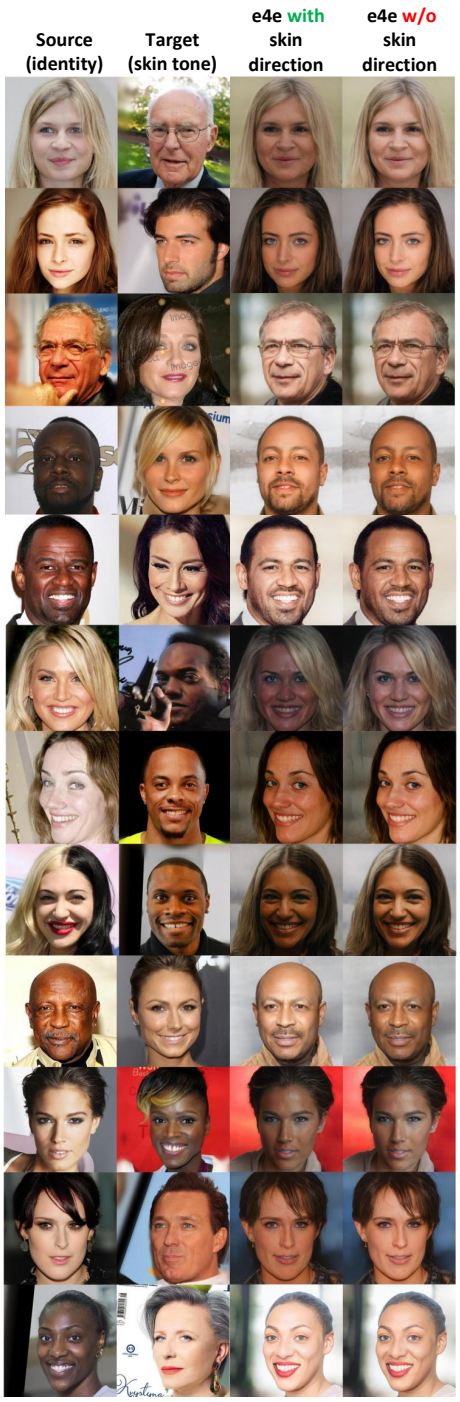


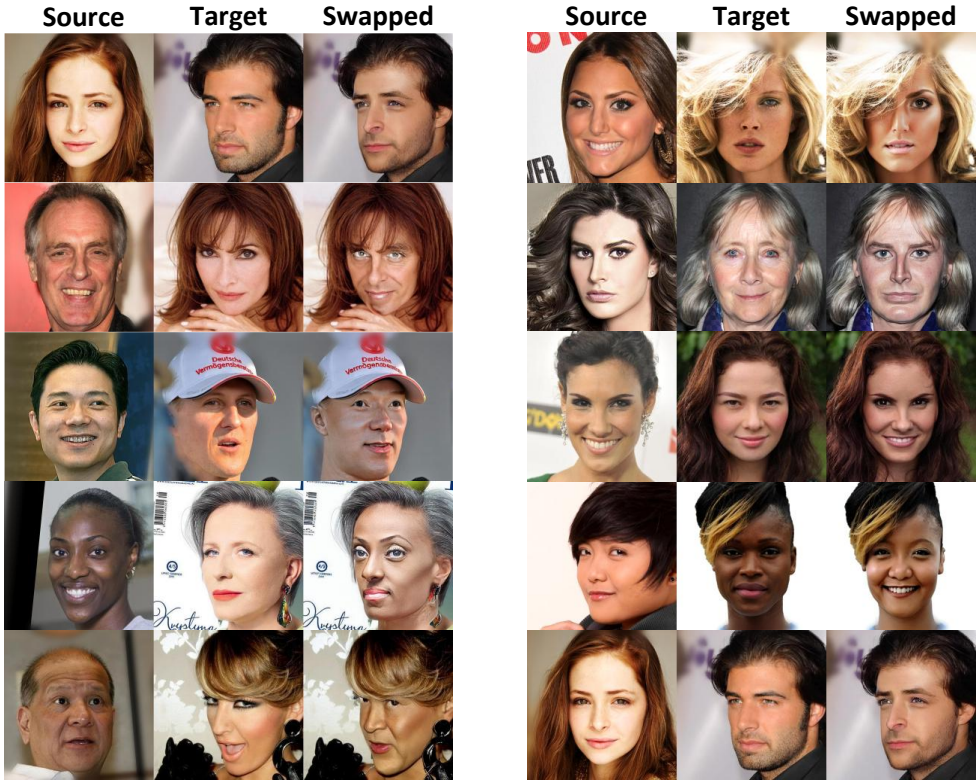
Figure 3.8: Qualitative result comparison between skin refinement module utilizing e4e encoder with and without skin direction approach.

appearance. For instance, when transferring the darker skin tone onto a lighter skinned source, the result is often lighter than the target. Conversely, when transferring a lighter skin tone

onto a darker skinned source, the result is often darker than the target. In contrast, with skin direction, the skin color of the generated face is perceptibly and faithfully matched to the target, creating much more coherent and realistic results.

In conclusion, the explicit manipulation along the skin direction vector successfully corrects the inaccuracies of the simple style mixing approach.

3.2.3. REFace tuning module



(a) (b)

Figure 3.9: Problems of the REFace mode, Identity preservation: The *swapped* column shows results where the identity is not fully preserved from the *source* and retains features from the *target*. (a) and Attribute controls: The *swapped* column shows results where the expression or pose differs from the *target* image (b).

While the REFace model provides a powerful foundation for face swapping, there are some limitations concerning **identity preservation** and **attribute control**, which motivate us to propose modifications to the training procedure to address them.

As can be seen in Fig. 3.9a, swapped results of the REFace frequently exhibit an *identity leak*, where features from the target person remains visible in the final outputs. The swapped face often appears to be a blend of the source and target identities rather than a clean transfer of the source's identity. To address this, our initial proposal is to enforce a stronger penalty for

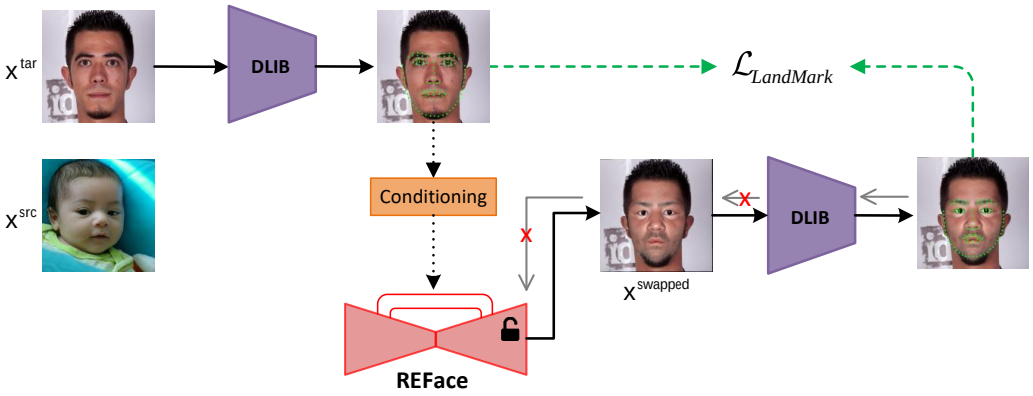


Figure 3.10: Pytorch cannot backpropagate through fixed array landmark coordinates predicted by DLIB library.

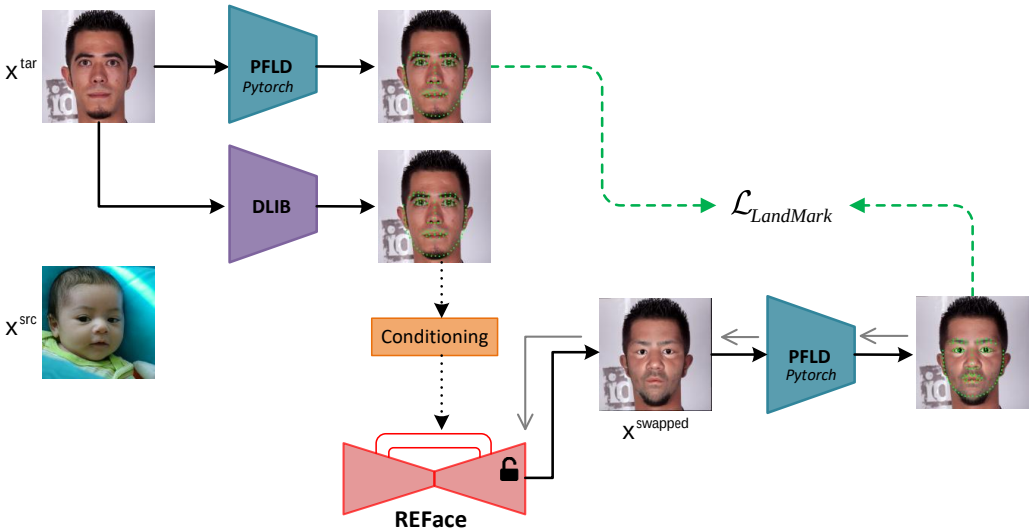


Figure 3.11: Proposal of using PFLD as landmark detector in implementing landmark loss function.

the identity deviation. We hypothesized that by increasing the weight of the identity loss in the total loss function, the model could be forced to prioritize a more accurate identity transfer. Therefore, we proposed increasing the ID loss weight (ω_{ID}) to **0.9**.

The second challenge is ensuring that other facial attributes from the target image, including expression and head pose, are preserved in the swapped face. Our observation in Fig. 3.9b shows that the baseline model does not always maintain the precise expression and pose of the target. This mismatch disturbs the coherence of the final results, as the generated faces are not photorealistic. To provide more explicit structural instruction, we introduced a **landmark loss function** ($\mathcal{L}_{landmark}$) to the training procedure. This loss penalizes deviation between the facial landmarks of the generated face and the target face. By adding this loss component, our goal is to force the expression and pose preservation. The total loss for the REFace training process in eq. (2.10) now becomes:

$$\mathcal{L}_{total} = \mathcal{L}_{Diff} + \omega'_{ID} \times \mathcal{L}_{ID} + \omega'_{LP} \times \mathcal{L}_{LP} + \omega'_{landmark} \times \mathcal{L}_{landmark} \tag{3.2}$$

where, $\omega'_{landmark}$ is corresponding weight for the landmark loss component.

Technical Challenge with a Naive Landmark Loss When we implement the landmark loss function, a significant technical challenge arises when using **DLIB library** [24], a standard and non-differentiable landmark predictor. The reason is that DLIB does not operate inside of the Pytorch computational graph and returns landmark coordinates as a *fixed Numpy array*. These arrays are static values and therefore they have no gradients with respect to parameters of the REFace model as demonstrated in Fig. 3.10. Consequently, the Pytorch framework cannot backpropagate through them and update the parameters.

To solve this issue, we employ a differentiable landmark loss detector that is compatible with the Pytorch framework, which named Pytorch-based Landmark Detector (**PFLD**) and illustrated in Fig. 3.11. This detector returns landmark coordinates as tensors that are part of the active computation graph. This allows gradients from the landmark loss to flow back to the REFace generator, enabling it to learn and correct errors in pose and expression.

4. Experiments

Contents

4.1. Datasets

Following the baseline approach [22], we conducted experiments on the **CelebAMask-HQ** dataset [4]. This is a large-scale and high-resolution dataset designed for computer vision tasks related to human face analysis. The key contribution of CelebAMask-HQ is the addition of annotated pixel level segmentation masks for every images compared to the original dataset - CelebA-HQ [10].

CelebAMask-HQ consists of 30,000 high-resolution face images, each having a resolution of 1024×1024 pixels. Corresponding to images, masks are annotated with 19 semantic classes. These classes cover all major facial features, providing data to train and evaluate sophisticated models for facial analysis, facial parsing, face editing, and manipulation,... Several images in this dataset can be seen in Fig. 4.1.

For the purpose of this project, we utilized a subset of the CelebAmask-HQ dataset, partitioned to manage the significant computational time required for each training experiments. Specifically, the dataset is used in three sets:

- **Training set**: 14,000 images, this part was used for the model's learning phase. During this stage, the model iteratively learned to generated swapped image which keeps the expression of target faces and preserves the identity of source faces.
- **Validation set**: 2,000 images, it was used to fairly evaluate performance of the current model during the training procedure, and guide hyperparameter optimization.

Figure 4.1: Several images in **CelebAMask-HQ** dataset [4].

- **Test set**: 1,000 images, this final set was completely unseen and is used to evaluate quantitative performance of the final model after completing one training experiment. In other words, it reflects the model's performance in a real-world scenario.

4.2. Evaluation Metrics

There are different dimensions to evaluate face swapping methods. In this thesis, we consider **identity preservation** between the swapped faces and the source faces, **realism and quality** of the swapped faces, and **fidelity** to measure the distortion between the swapped faces and the target ones.

ID feature similarity and ID retrieval For the identity preservation aspect, inspired by FaceShifter [28], we employ ArcFace [23] to extract ID features of source faces and swapped

results. These features are then used to compute cosine similarity by Eq. (4.1). A high similarity value (close to 1.0) indicates the identity is well preserved. In contrast, a score close to 0.0 signifies no similarity.

$$ID\_Feat\_Similarity = \frac{f_{src} \cdot f_{swapped}}{|f_{src}| \cdot |f_{swapped}|} \tag{4.1}$$

Apart from that, we search for the most similar face from all the source faces for each swapped result. Top-1 and Top-5 ranked are employed to measure ID retrieval score. Same as ID feature similarity, higher values are better.

Fréchet inception distance FID [29] is defined by the Fréchet distance between target feature vectors and swapped feature vectors extracted by Inception-v3 [30] pool3 layer. Unlike metrics comparing images pixel by pixel, FID uses high-level features and assesses the quality and diversity by comparing statistical properties between target faces and swapped results. A lower FID score indicates that the results are more similar to real images (target ones in this case) in terms of both realism and diversity.

Pose and Expression To evaluate the target attribute preservation, *pose* and *expression* errors are estimated by using HopeNet [31] and the 3DMM face reconstruction model [3]. To be more specific, the ℓ_2 distance of pose and expression between target images and swapped results is computed as metrics. Lower error values signify that the attributes are preserved well from target images to the results.

SSIM In face swapping problem, SSIM is useful for measuring the quality of the reconstructed areas and checking for the artifacts [32]. The principle of SSIM is that the human eye is highly adapted to extract structural information from a scene. Therefore, it should provides a good approximation of perceived image quality. SSIM compares target images and swapped ones by breaking the comparison into three components: **luminance, contrast** and **structure** with the following equation:

$$\text{SSIM}(I_{tar}, I_{swapped}) = \frac{(2 \times \mu_{tar} \times \mu_{swapped} + c_1) \times (2 \times \sigma_{tar,swapped} + c_2)}{(\mu_{tar}^2 + \mu_{swapped}^2 + c_1) \times (\sigma_{tar}^2 + \sigma_{swapped}^2 + c_2)} \tag{4.2}$$

μ_{tar} and $\mu_{swapped}$ are the pixel sample mean of target images and swapped images respectively; meanwhile σ_{tar}^2 and $\sigma_{swapped}^2$ are sample variance of target images and swapped images; $\sigma_{tar,swapped}$ is the sample covariance of the target and swapped images, while c_1 and c_2 are hyper-parameters. A high value of SSIM indicates that the swapped results have high fidelity and quality.

4.3. Implementation Details

4.3.1. Hyperparameters

Table 4.1 summarizes the different experimental scenarios used in our study. For each scenario, we vary the weights assigned to the identity loss (ω_{ID}), landmark loss ($\omega_{landmark}$), and LPIPS

Table 4.1: Weight of Loss functions Configurations for Different Experimental Scenarios.

scenario	ω_{ID}	ω_{LP}	$\omega_{landmark}$
baseline	0.3	0.1	0
our_I	0.9	0.1	0
our_II	0.3	0.1	0.3
our_III	0.9	0.1	0.9
our_IV	0.3	0.1	0.005

loss (ω_{LP}). The scenarios labeled **baseline** and **our_I–our_IV** correspond to different combinations of these hyperparameters to evaluate the impact of each loss component on the model's performance.

With each training experiment, the size of input images are 512×512 to adapt with computational resource. We set the batch size to 2 and the learning rate to 10^{-5}. We also employ the pretrained face recognition model of ArcFace [23] and the CLIP L/14 [33] foundation model the same as these models used in [22] as it is reliable for comparing our proposals with the original paper.

4.3.2. Software and Hardware Configurations

Our codebase is heavily borrowed from the baseline [22], We use Pytorch framework to implement the pipeline and conducted all experiments on 4 NVIDIA A100 GPUs(80GB) of the Austral Cluster of the CRIANN server, Centre Régional Informatique et d'Applications Numériques de Normandie, France.

4.4. Experimental Results

To assess the performance of our proposal pipeline, we conduct comprehensive experiments, comparing our whole pipeline with the original REFace baseline and several ablation studies. The results were evaluated using both quantitative metrics in section 4.2 and qualitative visual inspection.

4.4.1. Quantitative performance comparison

Our quantitative evaluation was performed in Tab. 4.2. For **FID**, **Pose**, **Expression**, lower is better. For all other metrics, higher is better.

The REFace baseline, which does not include a landmark loss, returns the best performance in terms of identity preservation, achieving the highest scores for *ID feature similarity* (**0.632**) and *ID retrieval* (**96.3%** and **98.6%**). In the meanwhile, applying our pipeline, which combines the lighting transfer module and skin refinement module, yields the best overall results in realism and attribute fidelity. Our proposal achieves *FID* of **7.160** and the best *Pose* error

Table 4.2: Quantitative result comparison between our proposal pipeline (**Ours**) and other experiments.

Methods	ID Feature Similarity ↑	ID Retrieval ↑		FID ↓	Pose ↓	Expression ↓	SSIM ↑
		Top 1 (%) ↑	Top 5 (%) ↑				
Baseline	**0.632**	**96.3**	**98.6**	7.435	3.184	0.95	0.743
ω_{id}=0.9	0.623	95.9	98.2	9.404	3.636	1.022	0.739
ω_{id}=0.3, $\omega_{landmark}$=0.3	0.471	74.4	85.2	21.843	5.222	1.200	0.717
ω_{id}=0.9, $\omega_{landmark}$=0.9	0.501	83	93	27.266	5.004	1.261	0.723
ω_{id}=0.3, $\omega_{landmark}$=0.005	0.626	**96.3**	**98.6**	8.592	3.411	1.012	0.744
Our	0.498	81.6	90.7	**7.160**	**3.089**	**0.942**	**0.676**

of **3.089**, the best *Expression* error of **0.9422**, and the highest $SSIM$ score of **0.767**. This demonstrates that while the baseline excels at identity, our whole pipeline provides a superior balance, considerably improving perceptual quality and coherence.

Regarding the impact of the **Landmark loss function**, our ablation study reveals that naively adding a landmark loss function with a high weight degrades the model's performance. With $\omega_{landmark}$ set to 0.9 or 0.3, the FID metric ruins significantly to **27.2** and **21.8**, respectively. To the best of my knowledge, the potential reason for this degradation might be that during the diffusion model's training, the output at early sampling steps is a noise image. Predicted landmarks on this noise would results in a heavily penalized loss without valuable information and perceptual artifacts.

These above analysis suggests that the landmark loss weight should be kept very small. By reducing $\omega_{landmark}$ to 0.005, the model's performance improves dramatically over the high weight values, but it still does not overcome the overall quality of the baseline.

4.4.2. Qualitative results comparison

A qualitative comparison is shown in Fig. 4.2, proving the findings from our quantitative analysis.

Impact of High Landmark Loss: The qualitative results illustrate the failure cases of high landmark loss weight. The images generated with $\omega_{landmark}$ set to 0.9 and 0.3 are disturbed by dramatic artifacts, color distortion, and unnatural facial structures. This provides strong visual confirmation for their poor FID score metric.

While the REFace baseline produces competent swaps, our method consistently generates more natural and coherent images. The key difference lies in the superior handling of lighting and skin tone, which are better matched to the target environment thanks to our preprocessing modules.

Our final proposed method produces the most visually appealing and realistic results. It suc-

Figure 4.2: Qualitative comparison between the whole pipeline (**Ours**) with the baseline model and other experiments.

cessfully transfers the source identity while faithfully preserving the target's expression, and pose. The seamless blending and absence of major artifacts highlight the performance of the proposal.

5. Conclusion

Our research has led to several key findings and contributions. Firstly, we confirm that a strong focus on identity loss, as can be seen in the REFace baseline, provides an excellent foundation for identity preservation. However, this alone is insufficient for achieving state-of-the-art realism. Our investigation reveals that a naive structural loss function such as a landmark loss with a high weight is inefficient for complex training process like latent diffusion models, as it can degrade image quality.

Secondly, the central contribution of this work is our three-stage pipeline that incorporating a lighting transfer module and skin refinement module, are crucial for generating coherent and realistic images. Our final model achieves superior attribute preservation metrics. This confirm our core hypothesis: better image quality is achieved by utilizing dedicated refinement modules that disentangle and solve specific sub-problems within the face swapping task.

Finally, while our proposal has demonstrated significant improvements, there are several promising directions for future works:

- Enhancing ID Conditioning: It could involve employing a more powerful **SwinFace** backbone, which has the potential to provide a richer identity representation, therefore possibly improving the fidelity of the identity transfer.
- Advanced Texture Conditioning: We could further leverage the 3DMM from [3] to extract a richer set of feature coefficients, including not only texture but also lighting and pose. Integrating these features into the conditioning vector could provide the model with more informative guidance.

Bibliography

[1] A. Hou, M. Sarkis, N. Bi, Y. Tong, and X. Liu, "Face relighting with geometrically consistent shadows," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4217–4226, 2022.

[2] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, "Encoding in style: a stylegan encoder for image-to-image translation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2287–2296, 2021.

[3] Y. Deng, J. Yang, S. Xu, D. Chen, Y. Jia, and X. Tong, "Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 0–0, 2019.

[4] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "Maskgan: Towards diverse and interactive facial image manipulation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5549–5558, 2020.

[5] C. Li, K. Zhou, and S. Lin, "Intrinsic face image decomposition with human face priors," in *European conference on computer vision*, pp. 218–233, Springer, 2014.

[6] A. Appel, "Some techniques for shading machine renderings of solids," in *Proceedings of the April 30–May 2, 1968, spring joint computer conference*, pp. 37–45, 1968.

[7] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, "Ganspace: Discovering interpretable gan controls," *Advances in neural information processing systems*, vol. 33, pp. 9841–9850, 2020.

[8] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

[9] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[10] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[11] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of gans for semantic face editing," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9243–9252, 2020.

[12] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or, "Designing an encoder for stylegan image manipulation," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–14, 2021.

[13] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

[14] P. Zhuang, O. Koyejo, and A. G. Schwing, "Enjoy your editing: Controllable gans for image editing via latent space navigation," *arXiv preprint arXiv:2102.01187*, 2021.

[15] A. Jahanian, L. Chai, and P. Isola, "On the" steerability" of generative adversarial networks," *arXiv preprint arXiv:1907.07171*, 2019.

[16] N. Spingarn-Eliezer, R. Banner, and T. Michaeli, "Gan" steerability" without optimization," *arXiv preprint arXiv:2012.05328*, 2020.

[17] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[18] Z. Wang, H. Zheng, P. He, W. Chen, and M. Zhou, "Diffusion-gan: Training gans with diffusion," *arXiv preprint arXiv:2206.02262*, 2022.

[19] Y. Chen, N. A. H. Haldar, N. Akhtar, and A. Mian, "Text-image guided diffusion model for generating deepfake celebrity interactions," in *2023 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 348–355, IEEE, 2023.

[20] C. Xu, S. Zhu, J. Zhu, T. Huang, J. Zhang, Y. Tai, and Y. Liu, "Multimodal-driven talking face generation via a unified diffusion-based generator," *arXiv preprint arXiv:2305.02594*, 2023.

[21] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

[22] S. Baliah, Q. Lin, S. Liao, X. Liang, and M. H. Khan, "Realistic and efficient face swapping: A unified approach with diffusion models," in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1062–1071, IEEE, 2025.

[23] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4690–4699, 2019.

[24] D. E. King, "Dlib-ml: A machine learning toolkit," *The Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[25] S. Czolbe, O. Krause, I. Cox, and C. Igel, "A loss function for generative neural networks based on watson's perceptual model," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2051–2061, 2020.

[26] E. H. Land and J. J. McCann, "Lightness and retinex theory," *Journal of the Optical society of America*, vol. 61, no. 1, pp. 1–11, 1971.

[27] T. M. Kodinariya, P. R. Makwana, *et al.*, "Review on determining number of cluster in k-means clustering," *International Journal*, vol. 1, no. 6, pp. 90–95, 2013.

[28] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen, "Faceshifter: Towards high fidelity and occlusion aware face swapping," *arXiv preprint arXiv:1912.13457*, 2019.

[29] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[31] N. Ruiz, E. Chong, and J. M. Rehg, "Fine-grained head pose estimation without keypoints," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 2074–2083, 2018.

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[33] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PmLR, 2021.