Fine-Grained Face Swapping via Regional GAN Inversion

Zhian Liu[1†] Maomao Li[2†] Yong Zhang[2†] Cairong Wang[3] Qi Zhang[2] Jue Wang[2] Yongwei Nie[1*]
[1] South China University of Technology [2]Tencent AI Lab
[3]Tsinghua Shenzhen International Graduate School

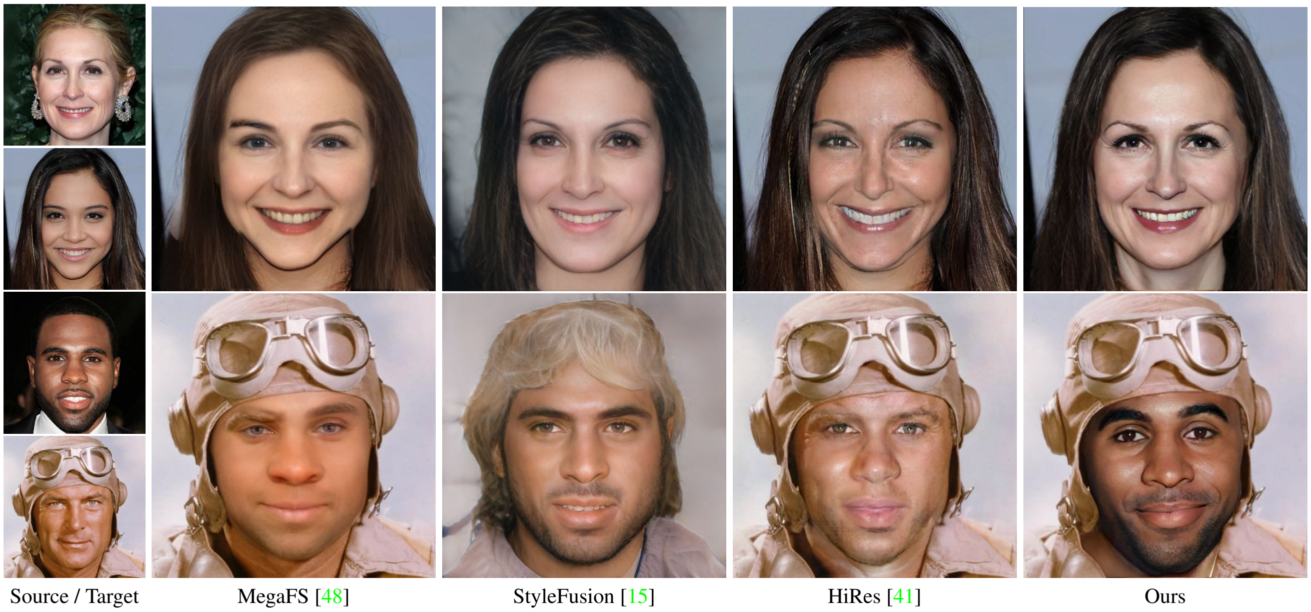


Figure 1. Compared with the existing StyleGAN-based face swapping approaches [15, 41, 48], our proposed method can achieve high-fidelity results that show better identity keeping from the source while keeping the similar pose and expression as the target. Note that skin color preservation and proper occlusion handling are our advantages over others. All the facial images are at 1024×1024.

Abstract

We present a novel paradigm for high-fidelity face swapping that faithfully preserves the desired subtle geometry and texture details. We rethink face swapping from the perspective of fine-grained face editing, i.e., "editing for swapping" (E4S), and propose a framework that is based on the explicit disentanglement of the shape and texture of facial components. Following the E4S principle, our framework enables both global and local swapping of facial features, as well as controlling the amount of partial swapping specified by the user. Furthermore, the E4S paradigm is inherently capable of handling facial occlusions by means of facial masks. At the core of our system lies a novel Regional GAN Inversion (RGI) method, which allows the explicit disentanglement of shape and texture. It also allows face swapping to be performed in the latent space of StyleGAN. Specifically, we design a multi-scale mask-guided encoder to project the texture of each facial component into regional style codes. We also design a mask-guided injection module to manipulate the feature maps with the style codes. Based on the disentanglement, face swapping is reformulated as a simplified problem of style and mask swapping. Extensive experiments and comparisons with current state-of-the-art methods demonstrate the superiority of our approach in preserving texture and shape details, as well as working with high resolution images. The project page is https://e4s2022.github.io

1. Introduction

Face swapping aims at transferring the identity information (*e.g.*, shape and texture of facial components) of a source face to a given target face, while retaining the identity-irrelevant attribute information of the target (*e.g.*, expression, head pose, background, etc.). It has immense potential applications in the entertainment and film production industry, and thus has drawn considerable attention in the field of computer vision and graphics.

The first and foremost challenge in face swapping is **identity preservation**, *i.e.*, how to faithfully preserve the unique facial characteristics of the source image. Most existing methods [9, 24, 38] rely on a pre-trained 2D face recognition network [12] or a 3D morphable face model (3DMM) [7, 13] to extract the global identity-related fea-

Work done when Zhian Liu was an intern at Tencent AI Lab
† Equal contribution.
∗ Corresponding author: nieyongwei@scut.edu.cn

tures, which are then injected into the face generation process. However, these face models are mainly designed for classification rather than generation, thus some informative and important visual details related to facial identity may not be captured. Furthermore, the 3D face model built from a single input image can hardly meet the requirement of robust and accurate facial shape recovery. Consequently, results from previous methods often exhibit the "in-between effect": *i.e*., the swapped face resembles both the source and the target faces, which looks like a third person instead of faithfully preserving the source identity. A related problem is **skin color**, where we argue that skin color is sometimes an important aspect of the source identity and should be preserved, while previous methods will always maintain the skin color of the target face, resulting in unrealistic results when swapping faces with distinct skin tones.

Another challenge is how to properly handle **facial occlusion.** In real applications, for example, it is a common situation that some face regions are occluded by hair in the input images. An ideal swapped result should maintain the hair from the target, meaning that the occluded part should be recovered in the source image. To handle occlusion, FSGAN [25] designs an inpainting sub-network to estimate the missing pixels of the source, but their inpainted faces are blurry. A refinement network is carefully designed in FaceShifter [24] to maintain the occluded region in the target; however, the refinement network may bring back some identity information of the target.

To address the above challenges more effectively, we rethink face swapping from a new perspective of fine-grained face editing, *i.e*., *"editing for swapping" (E4S)*. Given that both the shape and texture of individual facial components are correlated with facial identity, we consider to disentangle shape and texture explicitly for better identity preservation. Instead of using a face recognition model or 3DMMs to extract global identity features, inspired by fine-grained face editing [23], we exploit component masks for local feature extraction. With such disentanglement, face swapping can be transformed as the replacement of local shape and texture between two given faces. The locally-recomposed shapes and textures are then fed into a mask-guided generator to synthesize the final result. One additional advantage of our *E4S* framework is that the occlusion challenge can be naturally handled by the masks, as the current face parsing network [44] can provide the pixel-wise label of each face region. The generator can fill out the missing pixels with the swapped texture features adaptively according to those labels. It requires no additional effort to design a dedicated module as in previous methods [24, 25].

The key to our *E4S* is the disentanglement of shape and texture of facial components. Recently, StyleGAN [18] has been applied to various image synthesis tasks due to its amazing performance on high-quality image generation, which inspires us to exploit a pre-trained StyleGAN for the disentanglement. This is an ambitious goal because current GAN inversion methods [30, 33, 36] only focus on global attribute editing (age, gender, expression, etc.) in the global style space of StyleGAN, and provide no mechanism for local shape and texture editing.

To solve this, we propose a novel Regional GAN Inversion (RGI) method that resides in a new regional-wise $\mathcal{W}^+$ space, denoted as $\mathcal{W}^{r+}$. Specifically, we design a mask-guided multi-scale encoder to project an input face into the style space of StyleGAN. Each facial component has a set of style codes for different layers of the StyleGAN generator. We also design a mask-guided injection module that uses the style codes to manipulate the feature maps in the generator according to the given masks. In this way, the shape and texture of each facial component are fully disentangled, where the texture is represented by the style codes while the shape is by the masks. Moreover, this new inversion latent space supports the editing of each individual face component in shape and texture, enabling various applications such as face beautification, hairstyle transfer, and controlling the swapping extent of face swapping. To sum up, our contributions are:

- We tackle face swapping from a new perspective of fine-grained editing, *i.e*., *editing for swapping*, and propose a novel framework for high-fidelity face swapping with identity preservation and occlusion handling.
- We propose a StyleGAN-based Regional GAN Inversion (RGI) method that resides in a novel $\mathcal{W}^{r+}$ space, for the explicit disentanglement of shape and texture. It simplifies face swapping as the swapping of the corresponding style codes and masks.
- The extensive experiments on face swapping, face editing, and other extension tasks demonstrate the effectiveness of our E4S framework and RGI.

2. Related Work

GAN Inversion aims to map an image to its corresponding GAN latent code that can reconstruct the input as faithfully as possible. In this way, one can send the inverted-then-edited code to the generator to complete the expected editing. A number of StyleGAN inversion approaches have been proposed for face manipulation. Generally, they can be classified into three categories: (1) learning-based [3, 30, 34, 36, 42, 43], (2) optimization-based [1, 2, 16, 32, 46] and (3) hybrid methods [45]. The learning-based methods train an encoder to map the image to the latent space. In contrast, the optimization-based methods directly optimize the latent code to minimize the reconstruction error of the given image. The optimization-based approaches usually give better inversion performance but the learning-based methods cost less time. The hybrid methods make a trade-off between the above two, and use the inverted code as the starting point to conduct further optimization. Although specific face editing can be achieved by using existing GAN inversion methods, they work in a global fashion (e.g., growing old, pose changing, male to female) and cannot make precise control of the local facial component. Our Regional GAN Inversion fills in the gap of high-fidelity local editing via a novel $\mathcal{W}^{r+}$ latent space based on a pre-trained StyleGAN.

Face Swapping. The existing face swapping approaches can be generally classified into two categories [9], *i.e.*, source-oriented and target-oriented. The source-oriented approaches [5, 6, 25–27] start from the source and manage to transfer the attributes of the target to the source. The early methods in this camp can date back to [6], where 3D shape and relevant scene parameters were estimated to align pose and lighting. Then, [27] claimed that 3D shape estimation is unnecessary and proposed a face segmentation network to fulfill face swapping. Recently, a two-stage pipeline was introduced in FSGAN [25, 26], where a reenactment and an inpainting network tackle pose aligning and occlusion problems respectively. The target-oriented approaches [4, 9, 19, 22, 24, 38, 40] begin with the target and tend to transport the identity from the source. Generally, these technologies preserve the identity of the source by using a pre-trained face recognition model or 3DMMs. As the recognition model is trained for classification and 3DMMs are not accurate and robust, identity-related details cannot fully be captured for generation, bringing the “in-between effect”.

As for StyleGAN-based face swapping, MegaFS [48] applies the prior knowledge of pre-trained StyleGAN, raising the image resolution to 1024^2. StyleFusion [15] operates the latent fusion within the $\mathcal{S}$ space [10, 11], enabling controllable generation of local semantic region. However, the shape and texture of each facial region are still entangled in the $\mathcal{S}$ space. Beyond the global latent fusion, [40] designs a region-aware projector to transfer source identity to the target face adaptively. HiRes [41] employs an additional encoder-decoder for target features aggregation in a multi-scale manner. However, fine-grained and selective swapping is not supported in these two methods.

Our *E4S* belongs to the source-oriented camp. Inspired by mask-guided face editing [8, 23, 28, 47], we rethink face swapping from the perspective of face editing and treat it as editing of shape and texture for all facial components, *i.e.*, fine-grained face swapping. We propose to explicitly disentangle the shape and texture of facial components for better identity preservation based on the proposed RGI method, rather than using a face recognition model or 3DMMs.

3. Methodology

3.1. Editing-for-Swapping (E4S) Framework

Our *E4S* framework mainly consists of two phases inside: (a) reenactment, and (b) swapping and generation, where the overall pipeline is illustrated in Fig. 2.

Reenactment. We first crop the face region of the source image and target image, obtaining the cropped faces I_s and I_t. Then, we use the dlib [20] toolbox to crop the face region and detect the facial landmarks. Next, we follow the original StyleGAN [17] to align the cropped face and resize it to the resolution of 1024×1024.

In order to drive I_s to reach a similar pose and expression as I_t, we employ a pre-trained face reenactment model FaceVid2Vid [37], resulting in a driven face I_d. Such a face reenactment processing can be described as: $I_d = G_r(I_s, I_t)$, where G_r denotes the FaceVid2Vid model. Further, we estimate the segmentation masks M_t of the target face I_t and M_d of the driven face I_d, thus obtaining the target and driven pairs (I_t, M_t) and (I_d, M_d). For face parsing, we utilize an off-the-shell face parser [49], where each segmentation mask belongs to one of the 19 semantic categories. For simplicity, we aggregate the categories of symmetric facial components, resulting in 12 categories, *i.e.*, *background, eyebrows, eyes, nose, mouth, lips, face skin, neck, hair, ears, eyeglass, and ear rings*.

Swapping and Generation. In this phase, we would elaborate on the face swapping process in our E4S. We first feed the driven pair (I_d, M_d) and the target pair (I_t, M_t) into a mask-guided multi-scale encoder F_ϕ respectively, which extracts the style codes to represent the texture of each facial region. This step can be summarized as:

$$S_t = F_\phi(I_t, M_t), \quad S_d = F_\phi(I_d, M_d), \tag{1}$$

where S_t and S_d denote the extracted texture codes of the target and driven face, respectively. The detailed modules of the encoder F_ϕ are introduced in Sec. 3.2. Then, we exchange the texture codes of several facial components of S_t with those of S_d, obtaining the recomposed texture codes S_{swap}. Here, the swapped components are: *eyebrows, eyes, nose, mouth, lips, face skin, neck, and ears*. Note that the skin is carefully considered here since it is identity-related, while it is neglected by most existing works.

In addition to texture swapping, shape swapping is also required to realize the aim of face swapping. Since the shape is represented by facial masks, we start with an empty mask M_{swap} as a canvas and then complete the mask recomposition in the following steps. First, we keep the *neck* and the *background* layout of the target mask M_t and stitch their masks onto M_{swap}. Then, we stitch the inner face regions of the driven mask M_d, including *face skin, eyebrows, eyes, nose, lips, and mouth*. Finally, we stitch the *hair, eye glasses, ear, and ear rings* of the target mask M_t onto M_{swap}. Note that the driven mask M_d and the target mask M_t may not be aligned with each other perfectly, leading to some missing pixels in the swapped mask, which is always caused by the occlusion. We observe that these missing areas are usually between the facial skin and hair or between the facial skin and neck. As a solution, we fill up these

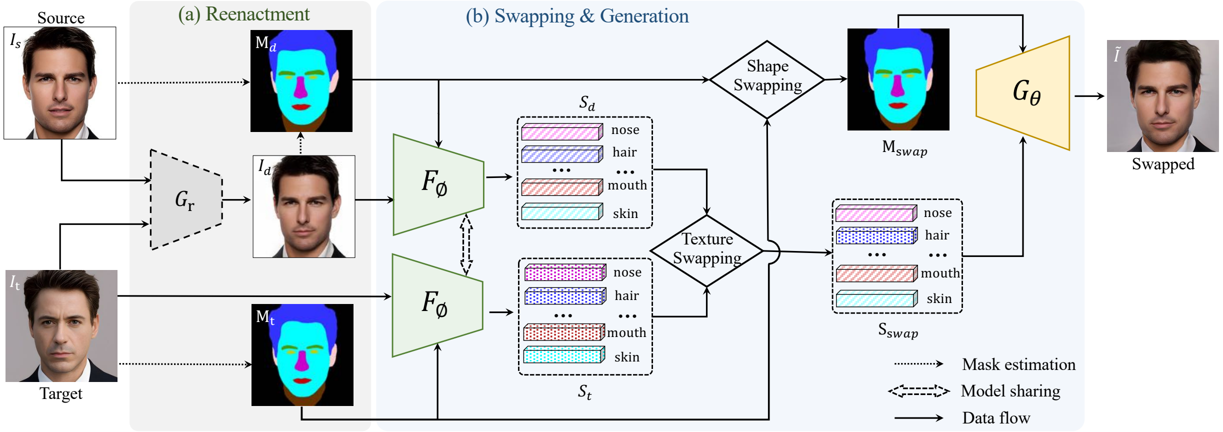


Figure 2. Overview of our proposed E4S framework. (a) For the source image I_s and the target I_t, a reenactment network G_r is used to drive I_s to show similar pose and expression towards I_t, obtaining I_d. The segmentation masks of I_t and I_d are also estimated. (b) The driven and target pairs (I_d, M_d) and (I_t, M_t) are fed into the mask-guided encoder F_ϕ to extract the per-region style codes to depict the texture respectively, producing texture codes S_d and S_t. We then swap the masks and the corresponding texture codes, and send them to the pre-trained StyleGAN generator G_θ with a mask-guided injection module to synthesize the swapped face $\tilde{I}$.

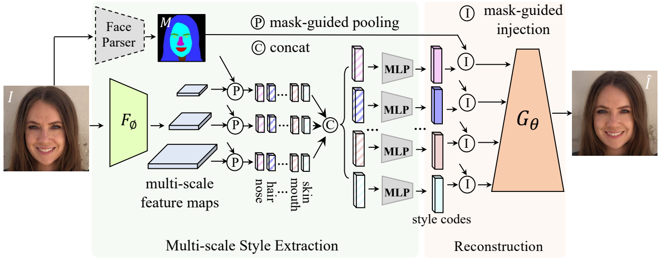


Figure 3. Overview of our proposed RGI. The input face I and its segmentation map M are fed into a multi-scale encoder F_ϕ to extract the per-region texture vectors. The multi-scale texture vectors are then concatenated and passed through some MLPs to obtain the style codes resident in a novel $\mathcal{W}^{r+}$ latent space of StyleGAN. The regional style codes and the mask M are used by our mask-guided StyleGAN generator to produce the reconstructed face $\hat{I}$.

areas with *face skin*, which is the unique advantage of our method. Compared with the existing methods FSGAN [25] or FaceShifter [24], our method *does not* need to train an extra sub-network to deal with the occlusion. Please refer to our Supp. for more details.

After obtaining the recomposed mask M_{swap} and texture codes S_{swap}, we feed them into the StyleGAN generator G_θ with a mask-guided style injection module to synthesize the swapped face, which can be expressed as $\tilde{I} = G_\theta(M_{\text{swap}}, S_{\text{swap}})$. Here, G_θ will be detailed in Sec. 3.2. Finally, the swapped face $\tilde{I}$ and target image T are blended together to produce the final swapped image.

3.2. Disentanglement of Shape and Texture

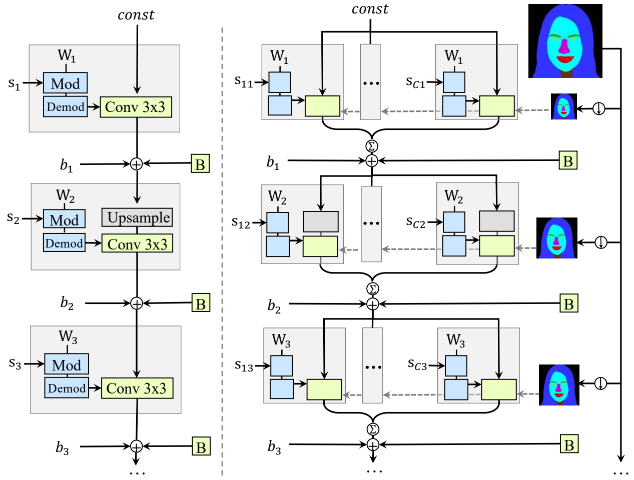
The core of our *E4S* framework is how to precisely encode the per-region textures which is disentangled with their shapes. Previous mask-guided face editing methods [23, 28, 47] attempt to use masks as input of a generator and inject the texture style to guide the generation, while they still struggle to preserve the identity and facial details during editing (see Fig. 8). Besides, they have a limited resolution of the generated face, where [23] reaches the resolution of 512^2 while the rest are with 256^2.

To pursue a better disentanglement of shape and texture as well as high-resolution and high-fidelity generation, we resort to the powerful generative model StyleGAN that can generate images with 1024^2 resolution. Instead of training StyleGAN from scratch, we explore the possibility of developing a GAN inversion method. Specifically, we use a pre-trained StyleGAN for the disentanglement, avoiding the massive computing resources and training instability. Although there are a number of GAN inversion techniques [30, 34, 43] have been proposed for face editing in the $\mathcal{W}$ or $\mathcal{W}^+$ space, they focus on global facial attribute editing only, *e.g.*, age, pose, and expression. Hence, they cannot be applied to the disentanglement of shape and texture for local editing. To tackle the shortage, we propose a novel Regional GAN Inversion (RGI) method for such a disentanglement, which incorporates facial masks into the style embedding and the generation process, thus filling in the gap of GAN inversion based local editing. The overview of our RGI is illustrated in Fig. 3.

Mask-guided Style Extraction. Given an image I and its corresponding segmentation mask M, we first feed the image I into a multi-scale encoder F_ϕ to produce feature maps $[F_1, F_2, ..., F_N]$ at different levels, where N is the number of scales and F_ϕ is a convolution network with multiple layers. Then, we can obtain the multi-scale features for each individual facial region based on the feature maps $[F_1, F_2, ..., F_N]$ and the mask M. Specifically, for each feature map F_i, we downsize the mask M to the same spatial size and apply the average pooling operation on F_i to aggregate features for each facial region as:

$$v_{ij} = \text{AVG}(F_i \odot (\lfloor M \rfloor_i == j)), \forall j \in \{1, 2, ..., C\}, \quad (2)$$

where C is the number of segmentation categories, $\odot$ is the



(a) Original StyleGAN (b) Mask-guided StyleGAN

Figure 4. The comparison of the original StyleGAN and the proposed mask-guided StyleGAN which regionally extends the style block. We sum up the intermediate feature maps of each region according to its segmentation mask that is downsized in advance.

Hadamard product, and $\lfloor M \rceil_i$ denotes the downsized mask with the same height and width as F_i. Further, the multi-scale feature vectors $\{v_{ij}\}_{i=1}^{N}$ of region j are concatenated and fed into an MLP to obtain the style codes:

$$s_j = \text{MLP}([v_{1j}; v_{2j}; ...; v_{Nj}]), \tag{3}$$

where s_j denotes the style codes of the j-th facial region. Then, the style codes and the mask M are fed into the StyleGAN generator to synthesize the swapped face. Here, we denote $s \in \mathbb{R}^{C\times 18\times 512}$ as the $\mathcal{W}^{r+}$ space.

Mask-guided Style Injection. As shown in Fig. 4(a), the original StyleGAN generator starts from a constant feature map with the spatial size of 4×4 and consists of a serials of style blocks. Each style block contains a modulation, a demodulation, and a 3×3 convolution layer. Besides, a noise layer $\boxed{\text{B}}$ is introduced to increase the diversity. The learnable kernel weights and bias in each block are denoted as W and b, respectively. W will be scaled by its corresponding style code with the shape of $\mathbb{R}^{512}$ before the convolution layer. An additional upsampling layer by the factor of two is employed between every two style blocks to increase the resolution of feature maps.

Different from the style code in the original StyleGAN, which globally controls the appearance of the output image, we propose to extract regional style code that controls only the appearance of the corresponding face component precisely along with its mask, as described above. To this end, we extend the style block of the original StyleGAN to a mask-guided style block conditioned on a given mask. Specifically, we sum up the intermediate feature maps with the guidance of per-region mask, which can be formed as:

$$F_l = \textstyle\sum_{j=1}^{C}(F_{l-1} * W'_{jl}) \odot (\lfloor M \rceil_l == j), \forall\, l \in \{1, 2, ..., K\}, \tag{4}$$

$$W'_{jl} = Demod(Mod(W_l, s_{jl})), \tag{5}$$

where F_{l-1} and F_l denote the input and output feature maps of l-th layer, respectively. W'_{jl} represents the scaled kernel weights for the j-th component in the l-th layer, and $*$ means the convolution operation. Similar to Eq. (2), $\lfloor M \rceil_l$ is the downsized mask corresponds to the l-th layer. We follow the same modulation and demodulation as the original StyleGAN and extend the style modulation regionally. In Eq. (5), W_l denotes the original kernel weights for the l-th layer, and the s_{jl} indicates the style code of j-th component for the l-th layer. The schematic operations of our proposed mask-guided style injection are illustrated in Fig. 4(b).

Note that the mask is only injected into the first K layers of the StyleGAN. That is, we do not use the mask-guided style block for the last $(18 - K)$ layers. There are two reasons for this occurrence: (1) we conduct experiments with $K = 11, 13, 15, 18$ and empirically find the reconstructed images show few visual differences when K is greater than 13; (2) the training overload can be decreased without the mask-guided style block in the last $(18 - K)$ layers since the resolution of these layers are large (*i.e.*, $512^2 - 1024^2$). Considering these two factors, we set $K = 13$ as the default in all the experiments.

3.3. Training Objective

During training, we only utilize the reconstruction as the proxy task and *do not* need to swap paired faces like the most existing face swapping methods, which makes our method more efficient and easier to train. Once the training is finished, the texture encoder F_ϕ can be used to produce per-region texture codes of any input face. One can easily achieve face swapping in the $\mathcal{W}^{r+}$ latent space as described in Sec. 3.1. We adopt the commonly used loss functions in the GAN inversion literature, which are described in our Supp. in detail.

4. Experimental setup

Datasets. **CelebAMask-HQ** [23] contains 30K high-quality face images, which are split into 28k and 2K for training and testing, respectively. This dataset also provides facial segmentation masks, with 19 semantic categories included. **FFHQ** [17] contains 70K high-quality images with a large diversity, but the facial segmentation masks are not officially given. We use a pre-trained face parser [49] to extract the facial segmentation masks.

Implementation Details. We use PyTorch [29] to implement our framework, and train our model on 8 NVIDIA A100 GPUs. During training, we set the batch size to 2 for each GPU and initialize the learning rate as 10^{-4} with the Adam [21] optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$). For CelebAMask-HQ and FFHQ datasets, we train the model for 200K and 300K iterations, respectively. The initial

Table 1. Quantitative comparison of our RGI under different ablative configurations. The reconstruction performance is measured.

Configurations	SSIM↑	PSNR↑	RMSE↓	FID↓
our RGI full model	0.818	19.851	0.105	15.032
(A) w/o identity loss	0.819	19.888	0.105	15.141
(B) w/o finetuning	0.827	19.984	0.104	22.239

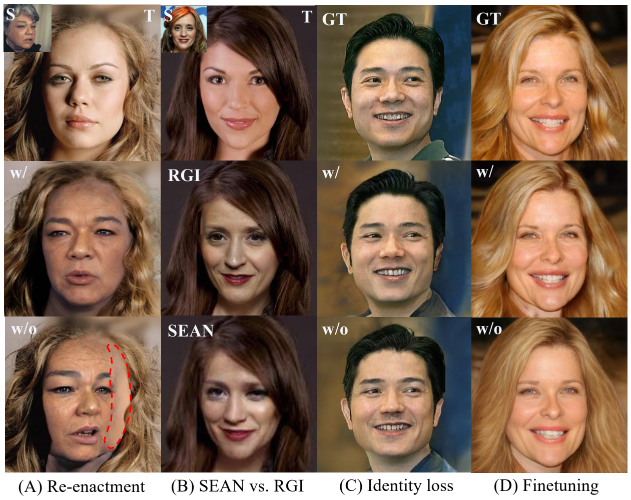


Figure 5. Qualitative comparisons of different ablative settings.

learning rate decays by the factor of 0.1 at 100K and 150K iterations, respectively. Besides, we randomly flip images with a ratio of 0.5.

5. Ablation study

In this section, we perform an ablation study to validate the design choices of our proposed *E4S* framework and RGI method. We show the qualitative comparison in Fig. 5 and the quantitative comparison in Tab. 1, where the reconstruction performance is considered.

Re-enactment. To drive the source to show a similar pose and expression as the target, we employ a pre-trained face reenactment model [37] before the shape and texture swapping procedure. To verify the necessity of the Re-enactment step in *E4S*, we compare a standard *E4S* pipeline and the one without Re-enactment. As shown in the 1st column of Fig. 5, the swapped result is not aligned with the target face when the reenactment is disabled (see the circled red region), revealing the pose information is also embedded in the per-region texture represented by the style codes.

SEAN *vs*. RGI. Our *E4S* framework is generic. Specifically, those methods which contain an encoder extracting the per-region style codes and a generator controlling the per-region style codes along with the segmentation mask, can be adapted to our *E4S* framework. To valid this, we replace our RGI with SEAN [47] to play the roles of F_ϕ and G_θ in Fig. 2. From the 2nd column of Fig. 5, we can observe that SEAN can produce an overall visually pleasant result while our result preserves more details (the eyes and face skin). Moreover, SEAN only shows the capability of generating faces at 256^2 while ours are at 1024^2. This also shows the superiority of our proposed RGI.

Identity loss. We add an ID loss when training our RGI under the reconstruction setting. From the configuration (A) in Tab. 1, the performance is comparable to the baseline when we do not apply the ID loss. However, without the ID loss would lead to some identity information missing, which is confirmed by the 3rd column in Fig. 5.

Pre-trained *vs*. fine-tuned StyleGAN. Though the pre-trained StyleGAN can be used for face swapping, the hair texture details cannot be always well preserved. To achieve a more robust performance on hair, we fine-tune the first $K = 13$ layers of the StyleGAN. The configuration (B) in Tab. 1 means we freeze the parameters of the StyleGAN generator and only train the texture encoder F_ϕ and the subsequent MLPs in our RGI. Although better SSIM, PSNR, and RMSE can be achieved by (B), the FID is poor. The last column in Fig. 5 illustrates an example. As shown, fine-tuning can improve hair quality while maintaining the texture of other inner facial components.

6. Face swapping results

We compare our method with the previous face swapping works: FSGAN [25], SimSwap [9], FaceShifter [24], and HifiFace [38]. We also compare with state-of-the-art StyleGAN-based face swapping methods, including MegaFS [48], StyleFusion [15], and HiRes [41]. Specifically, we train our model on the FFHQ dataset. Then, we randomly sample 500 source-target pairs from the CelebAMask-HQ and obtain the swapped results of each method.

Qualitative results. The qualitative comparisons are shown in Fig. 6 and Fig. 7. It can be observed that our method achieves more realistic and high-fidelity swapped results. Compared with FSGAN [25], our results are much sharper. For SimSwap [9] and Hififace [38], their swapped faces suffer from some artifacts and distortions (the 2nd row). Our E4S and FaceShifter [24] can generate visually satisfying results; however, our approach retains detailed textures better. We further compare the performance in more challenging cases where the occlusion exists in the source and target faces (the last two rows in Fig. 6). We can clearly see that our method can fill out the missing skin for the source face (the 3rd row), and maintain the glasses in the target face (the last row). Though a dedicated inpainting sub-network is designed in FSGAN, their inpainted results are quite blurry. FaceShifter proposes a refinement network to maintain the occlusion in the target image, but this may bring back some identity information of the target, making their swapped results similar to the target. Note that ours is the only method that can well preserve the skin color of the source, which is also an identity-related attribute. In case the source skin

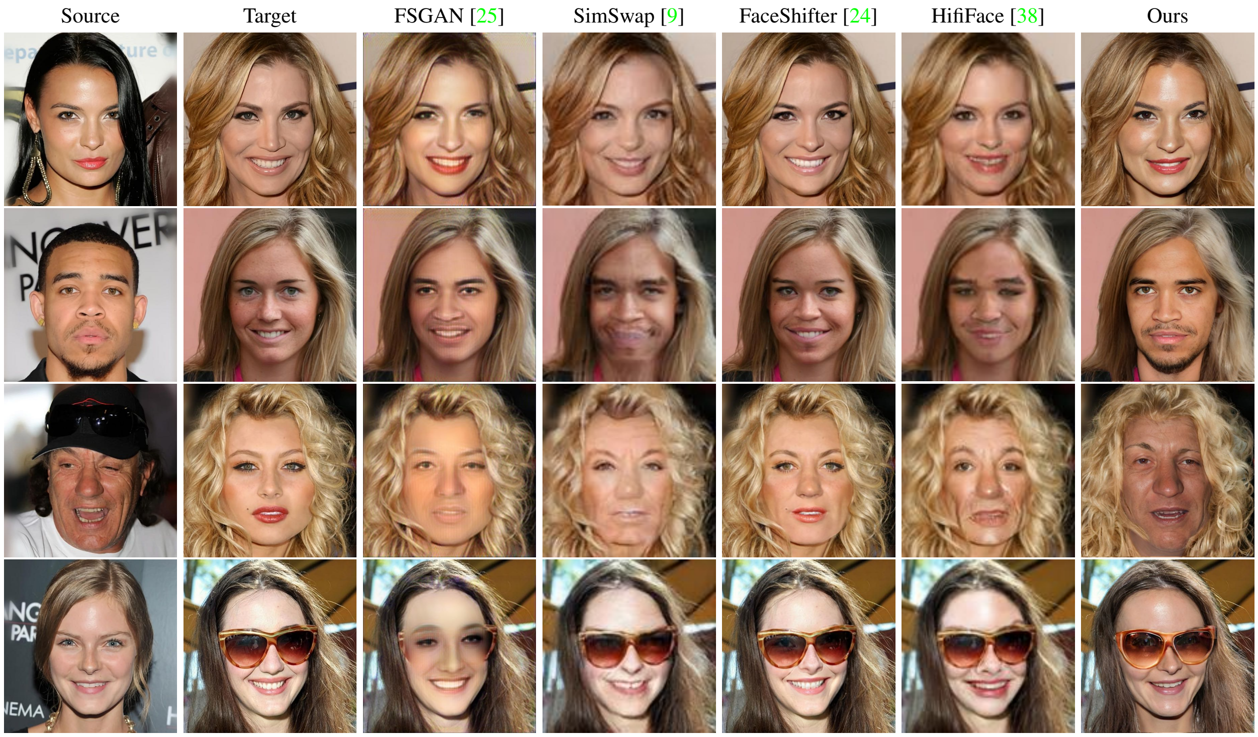


Figure 6. Qualitative comparisons of our results with state-of-the-art face swapping methods. Best viewed in color and zoom-in.

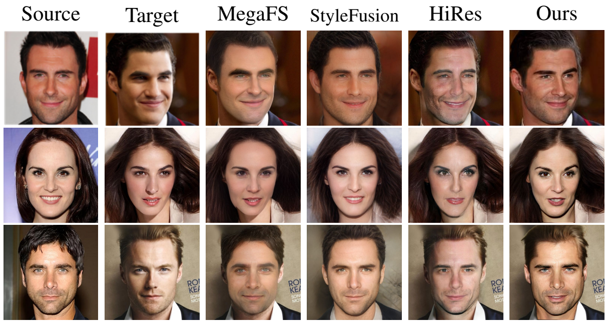


Figure 7. Comparisons with StyleGAN-based face swapping methods. Our method can achieve high-fidelity results while preserving the identity from the source better (*e.g.*, skin, beard, eyes).

is not fully adapted to the target due to the different lighting conditions, one can add a lighting transfer step for the source in advance. We leave disentangling the light from the texture in future work.

For a fair comparison, we also show the results of some StyleGAN-based approaches against ours in Fig. 7. Although all these methods utilize the pre-trained StyleGAN, we find the results of MegaFS [48] look to be a mixture of the source and target, which are blurry and lack of textures. The results of StyleFuison [15] show a bit of oversmoothing, *e.g.* the Adam Levine case. Though 1024^2 resolution can be achieved by HiRes [41], their results still suffer from some artifacts. In contrast, our method can generate more realistic and high-quality faces. Please refer to our supplement for more results.

Table 2. Quantitative comparison for face swapping. The numbers in **bold** denote the best results. †: source-oriented method, ‡: target-oriented method, *: StyleGAN-based method

Method	**ID retrieval↑**		**Pose↓**	**Expr.↓**
	Top-1	Top-5		
FSGAN † [25]	0.17	0.32	2.33	2.45
SimSwap ‡ [9]	0.12	0.32	2.89	2.84
FaceShifter ‡ [24]	0.06	0.26	**1.73**	**2.35**
HifiFace ‡ [38]	0.15	0.37	2.77	2.82
MegaFS * [48]	0.29	0.45	3.03	3.05
StyleFusion * [15]	0.35	0.18	5.37	2.94
HiRes ‡ * [41]	0.05	0.51	2.71	2.83
Ours † *	**0.38**	**0.54**	3.29	3.05

Quantitative results. We also conduct a quantitative comparison with the leading methods with respect to the identity preservation from the source and the attribute preservation from the target. The results are reported in Tab. 2. For source identity preservation, we first extract the ID feature vectors of all the source faces and the swapped results using CosFace [35]. For each swapped face, we perform face retrieval by searching for the most similar face from all the source faces. The similarity is measured by the cosine distance. Top-1 and Top-5 accuracy are the evaluation metrics. As for the target attribute preservation, we use HopeNet [31] and a 3D face reconstruction model [13] to estimate the pose and expression, respectively. We calculate the ℓ_2 distance of the pose and expression between each swapped face and its ground-truth target face.

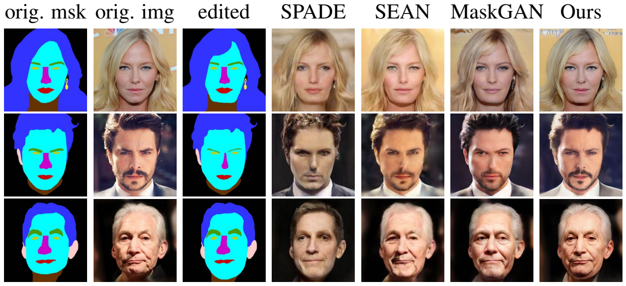


Figure 8. Qualitative comparisons of our results with state-of-the-art face editing methods. Some modifications are made on the face contour, hair, nose, and eyebrows. Our method can produce more high-fidelity editing results while maintaining the details of other components and the overall identity information well.

It can be observed that our method achieves the best retrieval accuracy, which demonstrates that our swapped faces keep the identity from the source mostly. The visual comparisons shown in Fig. 6 also support this observation. As for the target attribute preservation, our performance in pose and expression is still comparable with SOTA methods. Generally speaking, the target-oriented methods perform better in maintaining pose and expression, as they start from the target while our source-oriented method needs to generate these information starting from the source. However, one side effect of the target-oriented methods is that they only modify the shape and texture of the target face slightly and cannot fully preserve the identity (see FaceShifter in Fig. 6 and HiRes in Fig. 7). That is, there is a trade-off between identity and attribute preservation in these methods. Note that the accuracy of the face reenactment method in our *E4S* framework is the key factor that affects the pose and expression preservation. Our *E4S* framework is generic, and the performance could be further improved with a more advanced reenactment model.

7. Face editing results

Other than face swapping, our RGI resided in the $\mathcal{W}^{r+}$ space can also be used for fine-grained face editing conveniently. One can edit the mask of a specific region or swap the style of a specific region (*e.g.*, eyes, lips) with a reference image to obtain the desired editing results. This enables our RGI to support various applications such as face beautification, hairstyle transfer, and controlling the swapping extent of face swapping. Please consult our supplement for the details. In this section, we compare our method with the leading fine-grained face editing works: SPADE [28], SEAN [47], and MaskGAN [23]. For a fair comparison, we train our RGI network on the training set of CelebAMask-HQ and evaluate it on the test set. We use the officially released pre-trained models of the competing methods to obtain their inference results.

Qualitative results. We show the visual comparison with the competing methods in Fig. 8. We make some modifications to the original facial mask, such as hair, eyebrows, and chin. It can be observed that our approach produces more high-fidelity and natural editing results, where the details of other components and the overall identity information are well maintained.

Table 3. Quantitative comparison for image reconstruction on CelebAMask-HQ [23] test set. The rows in gray indicate the reconstruction images are obtained via style code optimization.

method	SSIM↑	PSNR↑	RMSE↓	FID↓
SPADE [28]	0.64	15.67	0.17	20.45
SEAN [47]	0.71	18.57	0.12	17.74
MaskGAN [23]	0.75	19.42	0.11	19.03
Our RGI	**0.82**	**19.85**	**0.10**	**15.03**
SofGAN [8]	0.76	14.86	0.19	26.73
RGI-Optim.	**0.86**	**23.02**	**0.07**	**14.73**

Quantitative results. We measure the image reconstruction quality of the competing methods and our RGI. The results are reported in Tab. 3, where the SSIM [39], PSNR, RMSE, and FID [14] are used as the metrics. We also compare with SofGAN [8], which is a StyleGAN-like generative model that relies on style code optimization for the reconstruction. For a fair comparison, an optimization stage is applied to our RGI (*i.e.*, RGI-Optim.). As shown in Tab. 3, our method always beats others in terms of all metrics, which indicates the visual inspection superiority of our method. We find SEAN [47] sometimes produces artifacts on hair regions. In contrast, our RGI can achieve high-fidelity reconstructions, keeping identity, texture, and illumination better. Besides, our RGI-Optim. can preserve the facial details better (*e.g.*, the curly degree of hair, the thickness of the beard, dimples, and background). For more visual comparisons, please check our supplement.

8. Conclusion

In this paper, we present a novel framework *E4S* for face swapping, which explicitly disentangles the shape and texture of each facial component and reformulates face swapping as a simplified problem of texture and shape swapping. To seek such disentanglement as well as high resolution and high fidelity, we propose a novel Regional GAN Inversion (RGI) method. Concretely, a multi-scale mask-guided encoder projects input faces into the per-region style codes. Besides, a mask-guided injection module uses the style codes to manipulate the feature maps in the generator according to the given masks. Extensive experiments on face swapping, face editing and other extended applications demonstrate the superiority of our method.

Acknowledgement

This research was sponsored by Natural Science Foundation of China (62072191). We thank Hanbang Liang for sharing the reproduced code of FaceShifter, and thank reviewers for their thoughtful comments and suggestions.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019. 2

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8296–8305, 2020. 2

[3] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6711–6720, 2021. 2

[4] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Towards open-set identity preserving face synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6713–6722, 2018. 3

[5] Dmitri Bitouk, Neeraj Kumar, Samreen Dhillon, Peter Belhumeur, and Shree K Nayar. Face swapping: automatically replacing faces in photographs. In *ACM SIGGRAPH 2008 papers*, pages 1–8. 2008. 3

[6] Volker Blanz, Kristina Scherbaum, Thomas Vetter, and Hans-Peter Seidel. Exchanging faces in images. In *Computer Graphics Forum*, volume 23, pages 669–676. Wiley Online Library, 2004. 3

[7] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999. 1

[8] Anpei Chen, Ruiyang Liu, Ling Xie, Zhang Chen, Hao Su, and Jingyi Yu. Sofgan: A portrait image generator with dynamic styling. *ACM transactions on graphics*, 2021. 3, 8

[9] Renwang Chen, Xuanhong Chen, Bingbing Ni, and Yanhao Ge. Simswap: An efficient framework for high fidelity face swapping. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2003–2011, 2020. 1, 3, 6, 7

[10] Min Jin Chong, Wen-Sheng Chu, Abhishek Kumar, and David Forsyth. Retrieve in style: Unsupervised facial feature transfer and retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3887–3896, 2021. 3

[11] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5771–5780, 2020. 3

[12] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019. 1

[13] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin 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*, pages 0–0, 2019. 1, 7

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 8

[15] Omer Kafri, Or Patashnik, Yuval Alaluf, and Daniel Cohen-Or. Stylefusion: A generative model for disentangling spatial segments. *arXiv preprint arXiv:2107.07437*, 2021. 1, 3, 6, 7

[16] Kyoungkook Kang, Seongtae Kim, and Sunghyun Cho. Gan inversion for out-of-range images with geometric transformations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13941–13949, 2021. 2

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 3, 5

[18] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 2

[19] Jiseob Kim, Jihoon Lee, and Byoung-Tak Zhang. Smoothswap: A simple enhancement for face-swapping with smoothness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10779–10788, 2022. 3

[20] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. 3

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[22] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. Fast face-swap using convolutional neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 3677–3685, 2017. 3

[23] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5549–5558, 2020. 2, 3, 4, 5, 8

[24] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Faceshifter: Towards high fidelity and occlusion aware face swapping. *arXiv preprint arXiv:1912.13457*, 2019. 1, 2, 3, 4, 6, 7

[25] Yuval Nirkin, Yosi Keller, and Tal Hassner. Fsgan: Subject agnostic face swapping and reenactment. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7184–7193, 2019. 2, 3, 4, 6, 7

[26] Yuval Nirkin, Yosi Keller, and Tal Hassner. Fsganv2: Improved subject agnostic face swapping and reenactment. *arXiv preprint arXiv:2202.12972*, 2022. 3

[27] Yuval Nirkin, Iacopo Masi, Anh Tran Tuan, Tal Hassner, and Gerard Medioni. On face segmentation, face swapping, and face perception. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 98–105. IEEE, 2018. 3

[28] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 3, 4, 8

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5

[30] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2287–2296, 2021. 2, 4

[31] Nataniel Ruiz, Eunji Chong, and James M. Rehg. Fine-grained head pose estimation without keypoints. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 7

[32] Rohit Saha, Brendan Duke, Florian Shkurti, Graham W Taylor, and Parham Aarabi. Loho: Latent optimization of hairstyles via orthogonalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1984–1993, 2021. 2

[33] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 2

[34] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021. 2, 4

[35] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018. 7

[36] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute editing. *arXiv preprint arXiv:2109.06590*, 2021. 2

[37] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10039–10049, 2021. 3, 6

[38] Yuhan Wang, Xu Chen, Junwei Zhu, Wenqing Chu, Ying Tai, Chengjie Wang, Jilin Li, Yongjian Wu, Feiyue Huang, and Rongrong Ji. Hififace: 3d shape and semantic prior guided high fidelity face swapping. *arXiv preprint arXiv:2106.09965*, 2021. 1, 3, 6, 7

[39] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 8

[40] Chao Xu, Jiangning Zhang, Miao Hua, Qian He, Zili Yi, and Yong Liu. Region-aware face swapping. *arXiv preprint arXiv:2203.04564*, 2022. 3

[41] Yangyang Xu, Bailin Deng, Junle Wang, Yanqing Jing, Jia Pan, and Shengfeng He. High-resolution face swapping via latent semantics disentanglement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7642–7651, 2022. 1, 3, 6, 7

[42] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. A latent transformer for disentangled face editing in images and videos. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13789–13798, 2021. 2

[43] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. Feature-style encoder for style-based gan inversion. *arXiv e-prints*, pages arXiv–2202, 2022. 2, 4

[44] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129(11):3051–3068, 2021. 2

[45] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European conference on computer vision*, pages 592–608. Springer, 2020. 2

[46] Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Barbershop: Gan-based image compositing using segmentation masks. *arXiv preprint arXiv:2106.01505*, 2021. 2

[47] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5104–5113, 2020. 3, 4, 6, 8

[48] Yuhao Zhu, Qi Li, Jian Wang, Cheng-Zhong Xu, and Zhenan Sun. One shot face swapping on megapixels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4834–4844, 2021. 1, 3, 6, 7

[49] zllrunning. face-parsing.pytorch. `https://github.com/zllrunning/face-parsing.PyTorch`, 2019. 3, 5