DeepFaceReshaping: Interactive Deep Face Reshaping via Landmark Manipulation

Shu-Yu Chen[1,†], Yue-Ren Jiang[1,2,†], Hongbo Fu[3], Xinyang Han[2], Zitao Liu[4,5], Rong Li[6], and Lin Gao[1,2,*]

[1]*Institute of Computing Technology, Chinese Academy of Sciences*
[2]*University of Chinese Academy of Sciences*
[3]*City University of Hong Kong*
[4]*TAL Education Group*
[5]*Guangdong Institute of Smart Education, Jinan University*
[6]*Zhejiang Lab Nanhu Headquarters*
[†]*Indicate Equal contribution*
[*]*Corresponding author*

Abstract

Deep generative models allow the synthesis of realistic human faces from freehand sketches or semantic maps. However, while being flexible, sketches and semantic maps provide too much freedom for manipulation and are thus not very easy to control by novice users. In this work, we present DeepFaceReshaping, a novel landmark-based deep generative framework for interactive face reshaping. To realistically edit the shape of a face by manipulating a small number of face landmarks, we employ neural shape deformation to reshape individual face components. We then propose a novel Transformer-based partial refinement network to synthesize the reshaped face components conditioned on the edited landmarks, and fuse the components to generate the entire face in a local-to-global approach. In this way, we limit possible reshaping effects within a feasible component-based face space. Our interface is thus intuitive even for novice users, as confirmed by a user study. Our experiments show that our method outperforms a traditional warping-based approach and the recent deep generative techniques.

Keywords: Face Reshaping, Deep Generative Model, Interactive Editing.

1. Introduction

Facial image editing is an important task of great interests in computer vision and computer graphics and with various applications in mass media and film industry. The recent interactive face image editing techniques can be roughly categorized into two groups from the perspective of image generation: full generation from conditional inputs [11, 24, 54] and partial manipulation based on image completion [32, 20]. While they achieve impressive results, these methods all require users to provide quality inputs similar to edge or semantic maps of real images. Sketches or semantic maps are flexible but provide too much degree of freedom for manipulation. For example, poorly drawn sketches would easily lead to unsatisfactory results. Such tools are thus not very friendly for users with no or little drawing skill.

On the other hand, previous studies have explored the parametric space of 3D human faces for various applications [4, 26]. The underlying space of faces can help tolerate errors of input sketches for sketch-based face image synthesis [6] or 3D face modeling [12]. Requiring casual inputs only is a nice feature when users have a rough idea of a target result. However, it also means that precise control is not very easy with such tools. This motivates us to require users to give minimal but accurate inputs and rely on the underlying shape space of faces to complete desired effects.

In this work, we present *DeepFaceReshaping*, a novel deep generative framework for generating desired reshaping effects of an input face image by manipulating only a small number of facial landmarks (Fig. 1). A possible solution to this problem is to first reconstruct a 3D face by using a global 3D morphable face model [4], then perform handle-based 3D deformation of the reconstructed 3D face, and finally warp the input image guided by the deformed

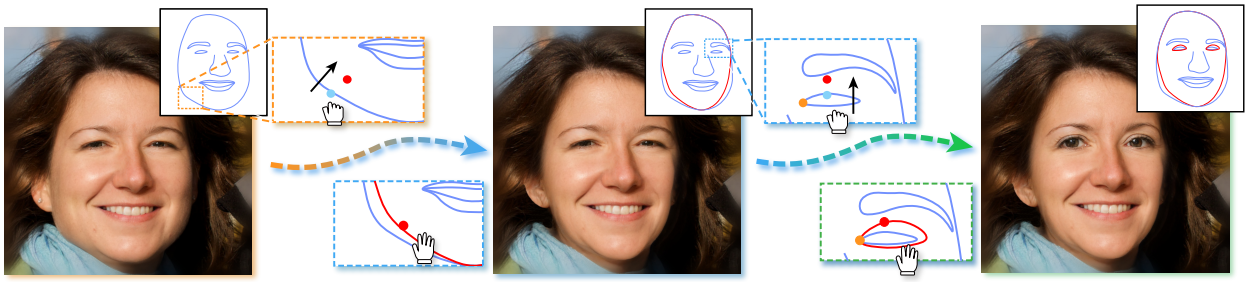


Figure 1: With our DeepFaceReshaping, novice users can manipulate a small number of facial landmarks to easily get realistic reshaping effects (Middle and Right) of an input face image (Left). Our generative model can synthesize plausible details with respect to user edits. The original landmarks are marked in blue, with the deformed landmarks overlapped in red. The landmarks before and after manipulation are highlighted in the closeups, with the black arrows showing the movement of the landmarks being manipulated.

3D face, similar to the pipeline for parametric reshaping of human bodies in images [53]. We show that it is possible to build similar morphable models in the 2D domain when a viewpoint is fixed (the frontal faces in our experiments), without using the complicated 2D-3D-2D pipeline. In addition, since relying on image warping is too limited to achieve various effects (e.g., to open a closed mouth in Fig. 1 (Middle)), generative models might be more suitable for image synthesis.

With the above key observations, we design a two-step approach: landmark-based neural shape deformation and local-to-global image synthesis, as illustrated in Fig. 3. In the first step, we take 2D facial landmarks of real faces in a dataset as driving examples. Since there exist natural correspondences between semantic landmarks from different faces, the corresponding landmarks implicitly define feasible deformation spaces. Given an input image for editing, we extract its landmarks and apply neural shape deformation modified from the method by Litany et al. [27] to update the remaining landmarks with respect to a small number of user-manipulated landmarks. In the second step, inspired by Chen et al. [6], we adopt a local-to-global generative network, which first synthesizes facial components according to the updated landmarks and then fuses the synthesized components and the background content into a complete face. We design a self-attention appearance encoder based on Transformer modules to encode the input image for preserving the facial attributes and identity.

In summary, our work makes the following two contributions:

- We develop a neural shape deformation method for interactive shape editing of face images. It not only supports an intuitive dragging-based interface but also guarantees the naturalness of facial shape.
- We propose a novel Transformer-based partial refinement network for the local-to-global network architecture to convert landmarks to realistic facial images. Our approach achieves clear improvements compared to the state-of-the-art conditional face generation methods.

2. Related Work

In this section we discuss the related works for interactive face generation and editing.

2.1. Face Generation with Deep Generative Models

In recent years, Generative Adversarial Networks (GANs) [10] have achieved impressive results in image generation, especially conditional face generation. Based on conditional GANs [28], Pix2Pix [18] and Pix2PixHD [43] were pioneering frameworks trained on paired data and solved various image-to-image translation problems. Since then such generative backbones were extended for various tasks [29, 24, 54, 42, 41, 6]. For example, Sangkloy et al. [35] used hand-drawn sketches and user-specified sparse color strokes as input to control properties of generated results. However, due to the data-driven nature, their framework tends to overfit to the edge maps and requires high quality of test sketches. To address this issue, DeepFaceDrawing [6] allowed the generation of realistic face images even from rough or incomplete sketches by projecting input sketches to the underlying manifolds of face components. Due to manifold mapping, DeepFaceDrawing does not provide precise control of synthesized results. In contrast, we aim to provide a novel face editing interface with intuitive and precise control.

To further control the styles and details of face generation, plenty of works have been done to explore the style space of human face. Most notably, Karras et al. [21, 22] introduced a style-based architecture as StyleGAN, which has a disentangled latent space showing great control capability of style. SPADE [29] used spatially-adaptive normalization

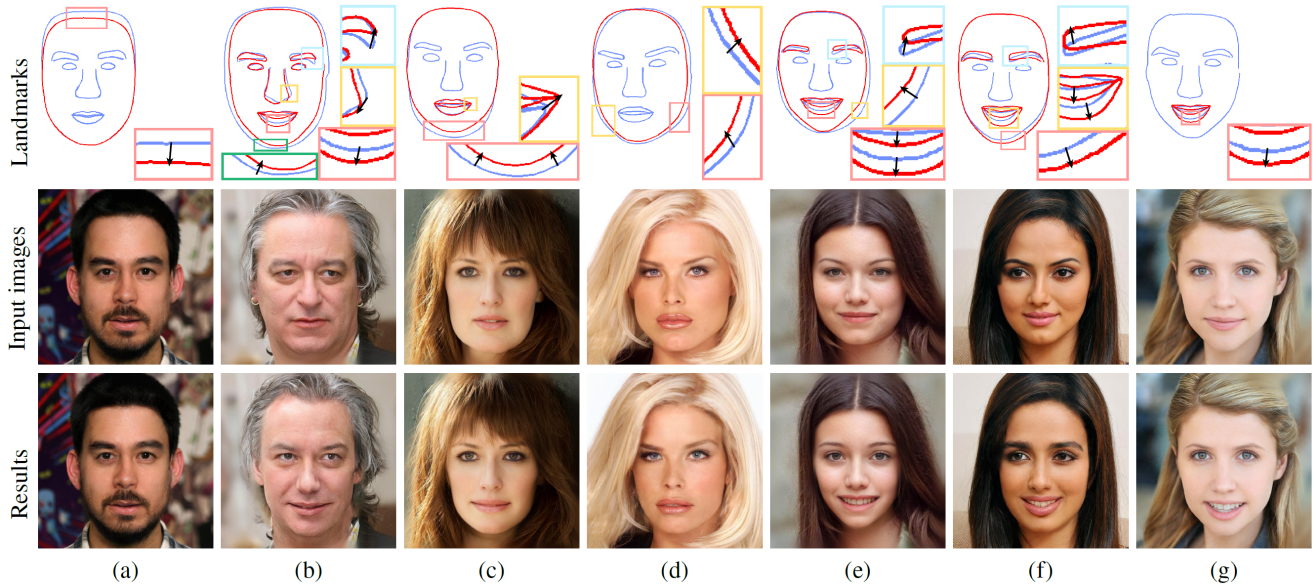


Figure 2: Editing results of our method, which can be used to achieve various effects such as lowering hairline (a), face reshaping (b, c, d, e), and smile adjusting (e, f, g).

to control over both semantics and style to enhance the semantic information of an input layout. SEAN [54] further improved SPADE by introducing per-region style encoding, which achieves style control of facial images at the level of segmentation masks. Their strategies of style control and image generation have shown great potential in deep image synthesis and are exploited by our framework.

2.2. Transformer Networks for Face Generation

Based on a self-attention mechanism, the original Transformer [40] was designed for natural language processing (NLP) tasks. Inspired by the ability of Transformer to capture the long-term correlation between complex sequential inputs, researchers have recently successfully applied Transformer to various computer vision tasks, including face image generation [30, 19, 8, 17].For example, Esser et al. [8] used an autoregressive transformer architecture to model the codebook learned by a CNN encoder for high-resolution image synthesis. Jiang et al. [19] built a GAN completely free of convolutions, using only pure transformer-based architectures. These works are pioneers to incorporate Transformer in image generation and inspired us to extend such a Transformer architecture in our generative networks for face image editing.

2.3. Neural Face Editing

Face editing and synthesis have been an active research topic in computer graphics and computer vision. From a historical perspective, the existing face editing methods can be divided into two major classes: classic image processing algorithms and deep generative methods. Traditional approaches often rely on image warping and texture rendering [25, 39, 23, 46]. For example, Averbuch-Elor et al. [3] animated a still face image by using confidence-aware warping and adding details and hidden regions from a driving video to the input image. Our approach does not require a driving video as input and focuses on interactive editing of a single image instead of video-to-image transfer. Han et al. [13] and Zhao et al. [52] both performed deformations on 3D reconstructed meshes and re-rendered the deformed textured meshes to obtain caricatures. In contrast, our goal is to synthesize realistic faces with respect to user-manipulated landmarks.

The recent deep generative models show a better ability to generate realistic images than traditional approaches. There are generally two categories of approaches for image editing through generative networks: one is to control the latent code in generative networks and the other is to edit the input of conditional networks. For example, Xiao et al. [47] proposed a model to exchange the latent codes of two faces to transfer their face attributes. The style-based architecture StyleGAN mentioned in Section 2.1 leads to many follow-up works (e.g., [37, 33, 31, 2]). However, these methods operate on a pre-defined set of attributes and do not support direct and precise geometry-level controls.

In order to enable fine-grained user control, conditional inputs are needed. Portenier et al. [32] and Jo et al. [20] adopted an image-completion strategy to achieve sketch-guided local editing. To have an overall control of facial shape and style, Gu et al. [11] learned feature embeddings

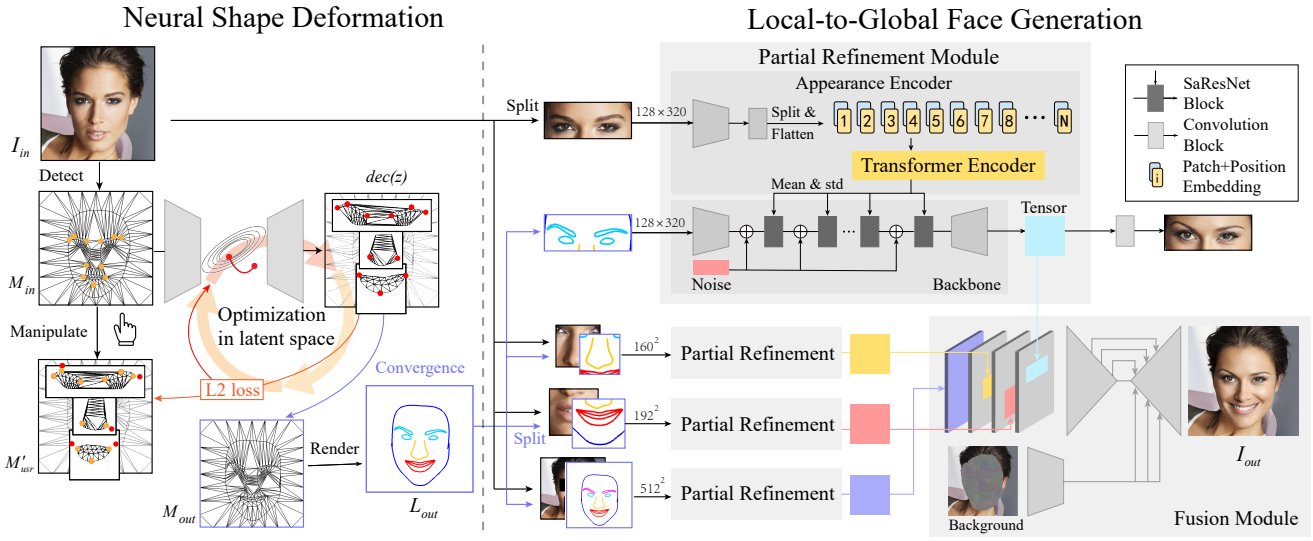


Figure 3: Illustration of the proposed DeepFaceReshaping framework. Given an input image I_{in}, its facial landmarks are first detected via a face alignment method and connected into a 2D triangle mesh M_{in}. The landmarks are divided into components and then updated by neural shape deformation with respect to the user-manipulated mesh M'_{usr} through optimization in the latent space, resulting in M_{out}, which is rendered as a landmark graph L_{out}. L_{out} is then split into component-level sub-graphs. The sub-graphs and the corresponding component regions in I_{in} are sent to the partial refinement modules (one for each component) for component-level feature embedding. For brevity, we show the detailed partial refinement module for the "eyebrows and eyes" component only. Finally, the global fusion module takes as input the concatenated partial refinement vectors of individual face components and generates a desired image I_{out} fused with the background.

for every face component of source images, and combined them with a target mask feature to generate a final output through a mask guided sub-network. Lee et al. [24] designed MaskGAN to learn style mapping and translated semantic masks into face images. Yang et al. [48] presented a sketch refinement strategy to handle face editing using sketch. Chen et al. [5] presented a structured disentanglement framework for sketch-based face editing and achieved realistic editing effects. Although these methods have accomplished conditional editing, the quality of results is still highly dependent on the quality of input sketches or semantic maps as well as the drawing skills of users.

3. Methodology

In this section, we describe our approach for landmark-based interactive deep face reshaping, which consists of two steps, namely, neural shape deformation and local-to-global face generation, as illustrated in Fig. 3. In the first step, our system first detects the facial landmarks, denoted as P_{in}, for a given face image I_{in} and connects the detected landmarks to form a 2D triangle mesh M_{in}, which consists of multiple semantic polygons corresponding individual face components. The mesh M_{in} is then encoded by a graph-convolutional variational autoencoder (VAE). We iteratively perform latent optimization to align the individual face components with the user-manipulated mesh M'_{usr} component by component. The deformed components are combined into the corresponding triangle mesh M_{out}, which is rendered as a landmark graph. In the second step, I_{in} and L_{out} are first decomposed into facial components, then processed by the corresponding partial refinement modules, and finally fused by a global fusion module to generate a desired image I_{out}. Ideally, I_{out} should have the same facial appearance and identity as I_{in} and respect L_{out}.

3.1. Neural Shape Deformation

The landmarks of a facial image define the geometric shape of the face, and the landmark collection from a dataset of faces defines the plausible face deformation space. Interacting with these landmarks in this deformation space is intuitive to explore the valid face geometric variations. We formulate the deformation in the 2D space, instead of going through a more complicated 2D-3D-2D process [12, 13]. To naturally support local edits and capture the detailed geometry variations, we propose to use a component-level deformation strategy. Specifically, we decompose the landmarks of a face into four components with connected meshes:

"face outline", "eyebrows and eyes", "nose", and "mouth", denoted as $e = \{1, 2, 3, 4\}$. We perform neural shape deformation component by component and use the user-manipulated landmark(s) belonging to a specific component as the constraint(s) to deform that component. It should be noted that the "face outline" is highly affected by the changes of internal components, so we calculate this part based on the landmarks of the entire face.

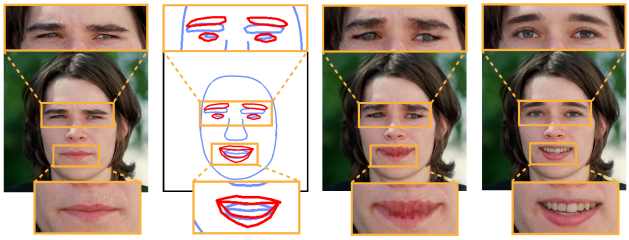
For intuitive landmark-based face editing, we employ a neural shape deformation method by Litany et al. [27], which originally operates on 3D meshes to solve the problem of shape completion. We adapt this optimization-based method to the 2D domain with a 2D triangle mesh. We first calculate the mean landmarks of our face dataset as the template where a Delaunay triangulation is generated to determine the graph connection and is used to form meshes. We then train a graph-convolutional VAE on the mesh collection of our face dataset to create the latent space, which parameterizes the embedding of the natural face shapes. At inference, the user-manipulated landmarks are given as a partially missing mesh M'_{usr}. The original mesh M_{in} is encoded for initialization, and the optimization of the latent code z is performed in the latent space to minimize the dissimilarity between the user-defined mesh M'_{usr} and the counterpart in the generated output shape:

$$\arg\min_{z \in space} ||dec(z)\Pi - M'_{usr}||_2, \quad (1)$$

where dec is the decoder for VAE and Π is a matrix to select the points in $dec(z)$ in the same indices of M'_{usr}.

3.2. Local-to-Global Face Generation

Given the deformed landmark graph, one direct way for transferring the deformation to the input image is to adopt image warping techniques, e.g., based on Moving Least Squares [36]. These methods can retain the information of the input images as much as possible, but they cannot synthesize new details that are missing in the input image (e.g., the teeth when the originally closed mouth is opened, as shown in Fig. 4). Inspired by recent image generation works [8, 6, 22], we tackle this problem by employing a local-to-global style-based image generation strategy instead of direct warping. As shown in Fig. 3, the input face image I_{in} and the landmark graphs after manipulation L_{out} are decomposed into four parts which are the same as the parts described in 3.1, and sent to the partial refinement modules to generate facial components separately. We then fuse them with the background condition into an entire image via a global fusion network. The outermost facial landmarks are used to mask out the frontal face region. To reconstruct the background region, we fill the foreground region with a random noise pattern in a similar way as [38] and encode the masked image for the fusion network.



(a) Original (b) Landmarks (c) MLS (d) Ours

Figure 4: Comparison between our deep generative method and an image warping approach based on moving least squares (MLS) [36]. (a) Input face image. (b) Landmarks before (in blue color) and after (in red color) manipulation. (c) Result of MLS [36] by using the deformed landmarks. (d) Our result. The superiority of our method is reflected in dealing with complex structures like the inner side of mouth with teeth and eyeballs.

3.2.1 Framework Architecture

Partial refinement modules. We design this module based on conditional GANs and Transformer encoders, as illustrated in Fig. 3. A special consideration we need to take is to keep the identity of the input face during generation, no matter how the landmarks are changed. We expect to train our network on static high-quality images without motion blur, which frequently occurs in video frames. Due to the lack of such data (i.e., paired images before and after manipulation), directly training a network to keep the identity is not feasible. We thus adopt a similar style control strategy used in many face image generation methods [24, 21, 50].

The partial refinement module consists of a backbone generation network and an appearance condition network. The input condition landmarks go through the backbone of down-sampling, SaResNet blocks, and up-sampling. SaResNet blocks are Resnet blocks that use Sandwich batch normalization [9]. To control the appearance and improve the quality of the generated faces, the partial refinement module must sufficiently learn the appearance information of the input face. Recent applications of Transformers [8, 19, 17] in image generation inspired us to utilize a Transformer to capture the high-level information of appearance, since they contain no inductive bias that prioritizes local interactions in contrast to CNNs. To develop a memory-friendly Transformer-based appearance encoder, instead of building on individual pixels, we first encode the input image through convolutional blocks into codes of high dimensions. The codes are split as a flattened sequence of $h \cdot w$ indices, where each patch of the sequence is treated as a "word" of style. The sequence is then combined with the learnable positional encoding and sent to a Transformer encoder to extract conditional information which is used in

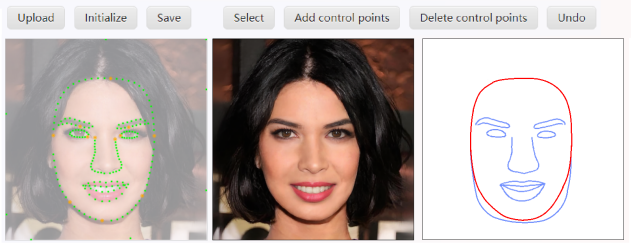


Figure 5: A screenshot of our landmark-based face reshaping interface. Users can select and drag the automatically detected facial landmarks on the left canvas and quickly get reshaping results in the middle.

the Sandwich batch normalization layers in the backbone. Inspired by StyleGAN [21] and SEAN [54], we also inject noise tensors scaled by learnable parameters to the input of ResNet blocks in the backbone for the enhancement of texture reconstruction.

Global fusion module. Given the embedded features of different components, there are various approaches for unifying these features into a realistic image. Our solution is to concatenate the partial features right before the last convolution layer of the up-sampling blocks of the partial refinement modules, since these features have the same shape and size as the input component images and can thus be concatenated directly according to the coordinates of the partial inputs without further alignment. To better maintain the contextual information of different components, this module is designed as a U-net [34] like structure. As illustrated is Fig. 3, given the embedding features with different shapes, we expand the features of smaller sizes to the same size by copying each feature to a zero tensor with the same shape as the input image. Then all the features are concatenated in the dimension channel and are convoluted to synthesize the final output with the encoded background image. The encoded background feature after each convolution layer has the same dimension as the input to the corresponding up-sampling layer and is merged progressively.

3.2.2 Training Strategy and Loss Functions

To train our network, we take a two-stage training strategy. First, the partial refinement networks are trained with the corresponding partial images. After the training of the local networks is converged, the embedded features are combined in the approach mentioned in Section 3.2.1 to train the global fusion network. To discriminate the output of the generator, we adopt the multi-scale discriminator used in Pix2PixHD [43]. We adopt the following loss terms to train the partial refinement and global fusing networks:

Adversarial loss: We use a two-scale PatchGAN discriminator D to match the distributions between generated results and real images in both the partial refinement and global fusion networks:

$$\begin{aligned}\mathcal{L}_{\mathbb{A}}(G, D) = \mathbb{E}[\log D(L_{in}, I_{in})] \\ + \mathbb{E}[1 - \log D(L_{in}, G(L_{in}, I_{in}))],\end{aligned} \tag{2}$$

where $D(L, I)$ is the output of discriminators and $G(L, I)$ is the output of generators.

Feature matching loss: To achieve more robust training of all the partial refinement modules and the global fusion network, we adopt the multi-scale discriminator feature matching loss used in Pix2PixHD [43]:

$$\mathcal{L}_{FM}(G, D_k) = \mathbb{E}\sum_{i=1}^{T}\frac{1}{N_i}[||D_k^{(i)}(L_{in}, I_{in}) - D_k^{(i)}(L_{in}, I_{out})||_1], \tag{3}$$

where T is the number of layers, N_i is the number of elements in the i-th layer, and k is the index of discriminators in the multi-scale architecture.

Lab color loss: To control the color tones of generated results, we measure the chromatic distance in a and b channels of images converted into the CIE LAB color space:

$$\mathcal{L}_c = ||\text{Lab}(I_{in})_{ab} - \text{Lab}(I_{out})_{ab}||_1. \tag{4}$$

Identity control: We calculate the cosine similarity between the ArcFace [7] embedding features of the input and output images to measure identity mismatch. Since these features are embedded by the whole face recognition network of ArcFace [7] and cannot be applied during the

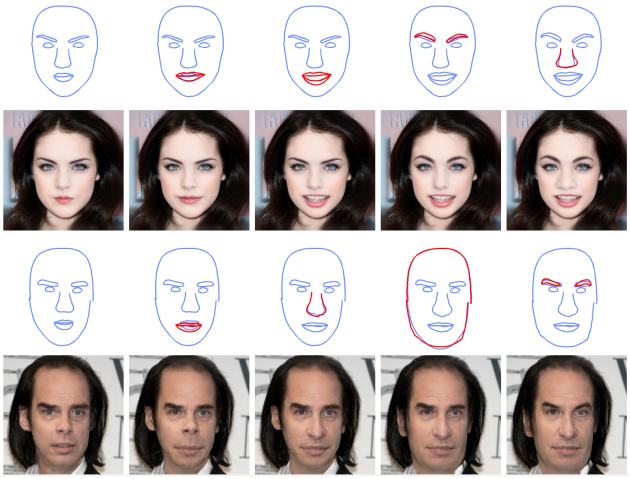


Figure 6: Two sequences of progressive (from left to right) editing and synthesis results. The first and third rows are the corresponding landmarks graphs. We highlight the recent changes of the landmarks in red. Our framework allows flexible editing of individual facial components.

discrimination of local face components, we also use the high-level feature loss with the pre-trained VGG19 model to boost local details in both the local and global training procedure, as shown in Eq. 6, where R stands for the pre-trained network ArcFace[7].

$$\mathcal{L}_{ID}^{local} = ||\text{VGG}(I_{in}^{c}) - \text{VGG}(I_{out}^{c})||_1 \quad (5)$$

$$\mathcal{L}_{ID}^{global} = ||\text{VGG}(I_{in}) - \text{VGG}(I_{out})||_1 + \lambda_{id}(1 - \langle \text{R}(I_{in}), \text{R}(I_{out}) \rangle). \quad (6)$$

Total loss: For the training of the partial refinement network, the total training loss is combined as: $\mathcal{L} = \mathcal{L}_{\mathbb{A}}(G, D) + \lambda_{FM}\mathcal{L}_{FM}(G, D_k) + \lambda_c\mathcal{L}_c + \lambda_{ID}\mathcal{L}_{ID}^{local}$, where $\lambda_{FM} = 10.0, \lambda_c = 1.0$, and $\lambda_{ID} = 10.0$. For the training of the global fusion network, the total training loss is combined as: $\mathcal{L} = \mathcal{L}_{\mathbb{A}}(G, D) + \lambda_{FM}\mathcal{L}_{FM}(G, D_k) + \lambda_c\mathcal{L}_c + \lambda_{ID}\mathcal{L}_{ID}^{global}$, where $\lambda_{FM} = 10.0, \lambda_c = 1.0$, and $\lambda_{ID} = 1.0$.

4. Experiments

In this section, we compare our proposed technique to the state-of-the-art methods, both quantitatively and qualitatively. We also present the results of an ablation study.

4.1. Setup

Dataset and data preparation. We evaluate our model on the CelebA-HQ dataset [45], which consists of 30K face images. First, we adopt the face alignment algorithm used in the data cleaning of the FFHQ dataset [21] to fix the locations of facial components and remove the faces with yaw angle out of the range $[-15°, +15°]$ to focus on the front view. Next we obtain dense facial landmarks of 772 points via Face++ Dense Facial Landmarks API [1]. Since the landmark labels cannot present the existence of eyeglasses, we exclude the face images with eyeglasses from the dataset. Finally, 18K images are remained and divided into a training set and a testing set at the ratio of 9:1. The landmarks are then connected as 2D triangle meshes and converted into semantic maps in the same form as CelebAMask-HQ [24]. To ensure a fair comparison with the other methods, all the training and testing images used for comparisons are resized to 256×256 resolution.

Implementation details. For the input image size of different facial components, in our following experiments we set 512×512, 128×320, 160×160, and 192×192 for "remainder", "eyebrows and eyes", "nose", and "mouth", respectively. As for networks settings, we use the Adam optimizer with a learning rate of 0.0001. We use instance normalization for down-sampling and up-sampling layers. Please refer to the supplementary material for more details.

For the deformation interface, We obtain dense facial landmarks of 772 points but in practice we only use 193 of them by selecting points at an interval of three for deformation and landmark graph rendering. This not only reduces the inference time for deformation but also facilitates manipulation for users. We show our interface in Fig. 5.

Existing methods for comparison. We compare our method with four state-of-the-art open-source methods: SEAN [54], CoCosNet [51], MaskGAN [24], and Bilayer [49], using their official implementations. We also compare an image warping approach based on moving least squares (MLS) [36]. The first three works take a reference image and a semantic mask as the conditional input of networks to manipulate the entire face image. We trained them all on the CelebA-HQ dataset [45], with the same semantic maps converted from our dense landmarks. Bilayer [49] is a kind of face reenactment method, which is originally designed to create head avatars from a single photograph but not to reshape faces. Since it also takes facial landmarks as input, we compare this method in the manner of visual results to show the difference between face reenactment and editing methods.

Evaluation metrics. We reconstruct the images in the test set and compare the results of the existing methods and ours under two different perspectives: 1) the structural similarity (SSIM) [45] and peak signal-to-noise ratio (PSNR) to evaluate the reconstruction quality; 2) the learned perceptual image patch similarity (LPIPS) to evaluate the perceptual similarities. If the reconstruction results cannot maintain the characteristics of the input images, it would be difficult to perform further editing. We thus use the reconstruction results to measure the performance of the compared generation networks.

We also evaluate all the methods on the edited images. For making a fair comparison, we choose to acquire reshaped landmarks by randomly reshaping a face image. We sample a latent vector in the latent space of our VAE (Section 3.1) and interpolate it with the latent vector encoded from a real face image to get the edited latent vector, which is then decoded to get the reshaped landmarks. The following metrics are used for this evaluation: 1) the Fréchet Inception Distance (FID) [14] to measure the diversity and quality of generated images; 2) point-wise accuracy (PwA) to measure the consistency between the input landmarks and the detected [1] landmarks of the generated faces; 3) face verification loss [16] (CurFR) to evaluate the performance on identity preservation, which is independent from our loss function.

4.2. Qualitative Results

We show progressive editing results in Fig. 6. Our method supports detailed local edits thanks to the partial refinement modules used in image generation. In Fig. 7, we show qualitative comparisons with the state-of-the-art methods. It can be observed that our results are more refined than those by the compared methods, especially in the regions of eyes and mouth. The added LAB loss $\mathcal{L}_c$ is able

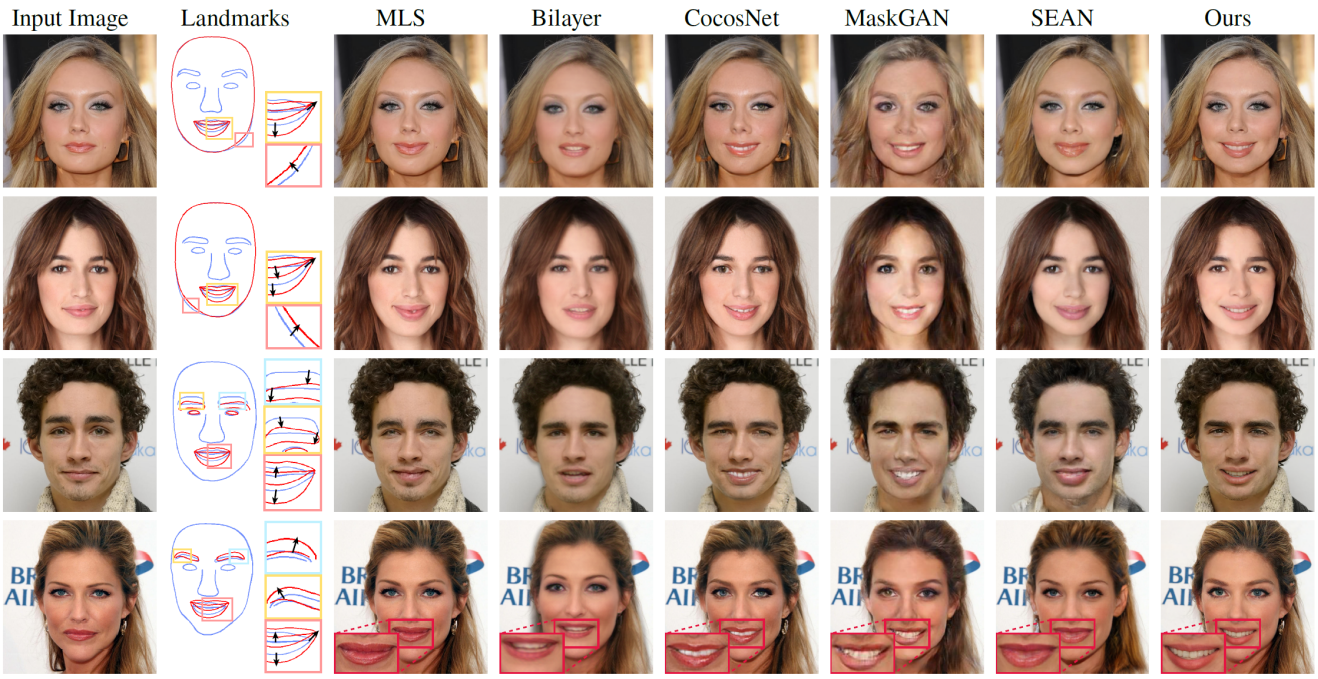


Figure 7: Comparisons with the state-of-the-art methods given the same deformed landmarks (the second column). For fair comparison, the background is added by one copy-paste-blend step for the results of all the face generation methods.

Table 1: Quantitative comparisons with the existing methods. Our method outperforms the compared methods for static facial image synthesis.

Method	SSIM↑	PSNR↑	LPIPS↓	FID↓	PwA↓	CurFR↑
SEAN	0.696±0.152	28.70±4.19	0.273±0.063	45.93	15.21±1.66	0.64±0.17
CocosNet	0.731±0.123	28.85±3.61	0.252±0.053	50.64	12.63±1.47	0.73±0.14
MaskGAN	0.679±0.164	28.79±4.26	0.301±0.077	61.31	12.88±1.73	0.61±0.18
Bilayer	0.630±0.136	28.65±4.17	0.309±0.058	54.55	17.92±1.54	0.67±0.15
Ours	**0.761±0.119**	**29.12±3.46**	**0.237±0.052**	**43.97**	**11.05±1.42**	**0.78±0.13**

to adjust the global hue to better match the input image than MaskGAN [24] and SEAN [54]. MLS [36] failed to synthesize the hidden regions such as inner mouth. The results of Bilayer [49] are realistic in the aspect of motion transfer but lack variations of shapes. CocosNet [51] retains most regions well but generates blurry teeth. This is possibly because CocosNet learns the correspondence between the image exemplar and input label but there is no exposed teeth in the image exemplar.

4.3. Quantitative Comparison

In Tab. 1, we report the quantitative evaluation in comparison with the state-of-the-art techniques. We exclude MLS [36] in this experiment since our purpose here is to compare the generative ability of deep learning models. It can be found that our method performs better than the compared methods. Based on visual perception of the results, we find that FID is more related to the integrity of generated faces while SSIM indicates the similarity between the reference images and the reconstructed images as expected. The difference in terms of PSNR is relatively minor, probably because our editing area only accounts for a small part of the whole figure while PSNR does not perform well in discriminating structural contents in images, as revealed in the previous studies [15, 44]. Due to the noise in the detection of landmarks, PwA has more noise than SSIM, LPIPS and CurFR, but the average values of the compared methods under PwA are still different.

4.4. Ablation Study

We perform an ablation study both quantitatively and qualitatively to verify the impact of individual components

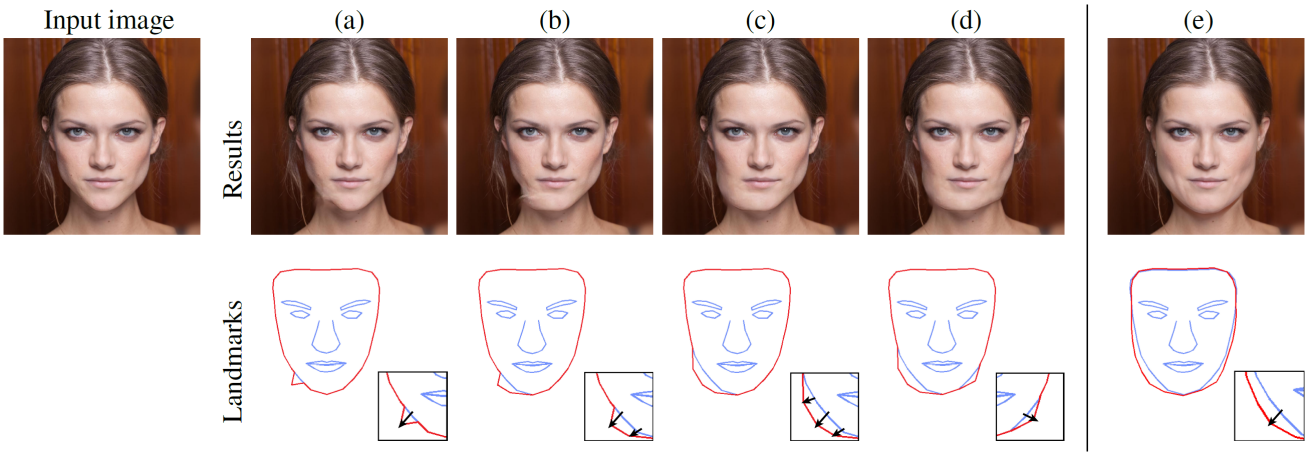


Figure 8: Editing results without (a-d) and with (e) landmark optimization. From (a) to (d), the user drags one to four points, respectively. We render L_{out} directly from M_{usr} without landmark optimization. In (e), the user drags only one point and we render L_{out} from M_{out} with landmark optimization.

Table 2: Ablation study on the CelebA-HQ dataset. Our full framework of the local-to-global network with noise injected leads to the best results. The confidence intervals with 0.95 confidence level are shown in the above table.

Setting	SSIM↑	PSNR↑	LPIPS↓	FID↓	PwA↓	CurFR↑
Global w/o $\mathcal{L}_c$	0.661±0.157	28.83±4.55	0.291±0.072	57.18	12.02±1.66	0.66±0.17
Global w/o noise	0.671±0.138	28.89±3.92	0.280±0.064	55.17	12.12±1.57	0.62±0.15
Global w/o Transformer	0.655±0.135	28.87±3.78	0.283±0.055	56.69	13.69±1.61	0.59±0.14
Global	0.676±0.142	28.91±4.40	0.268±0.068	47.20	11.98±1.52	0.68±0.15
Ours w/o $\mathcal{L}_{ID}$	0.694±0.123	28.96±3.62	0.257±0.053	46.09	11.52±1.46	0.57±0.16
Ours w/o optimization	**0.761±0.119**	**29.12±3.46**	**0.237±0.052**	51.32	14.94±1.49	0.63±0.18
Ours				**43.97**	**11.05±1.42**	**0.78±0.13**

in our model. We first perform the ablation comparison on landmark optimization to show how it can help reduce the number of user operations, as shown in Fig. 8. In (a)-(d), the results exhibit noticeable artifacts since the face outlines with the changes only caused by the user-edited landmarks are unrealistic. In contrast, the landmark optimization (e) adjusts all the landmarks along the face outline with respect to the user-edited landmarks. We add the ablation quantitative evaluation in Tab. 2, where "Ours w/o optimization" refers to the our results directly using M_{usr} (without the landmark optimization) to render semantic maps. As the reconstruction process does not involve optimization, "Ours w/o optimization" and "Ours" share the same results in SSIM, PSNR, and LPIPS. It can be seen that without this module our method leads to poorer performance in terms of FID, PwA, and CurFR.

We also perform the ablation comparison on different settings of generative networks to evaluate the effect of each module. As shown in Tab. 2, all the "Global" settings, i.e., without the partial refinement modules, lead to poorer performance than the local-to-global framework, denoted as "Ours". The method of injecting learnable noise also leads to higher performance. This is also confirmed by visual comparisons, as shown in Fig. 9. It is obvious that skin colors can be different without the Lab color loss $\mathcal{L}_c$ in Eq. 4, as confirmed by the heat maps. We also compare the performance of the appearance encoders with and without the Transformer encoder. It can be seen that the version without the Transformer design has lower scores. The added noise is useful to improve the synthesis of the detailed texture.

One crucial point of our framework is to maintain the identity while reshaping a face. Since we believe that the identity of a face is determined by both the texture and shape, our goal is more likely to keep the origin texture. The ability to control identity in our method is basically accomplished by the appearance encoder and is ensured by the identity losses during training. The visual comparison between the results with and without the identity control losses in Fig. 9 shows the positive effects of our identity control.

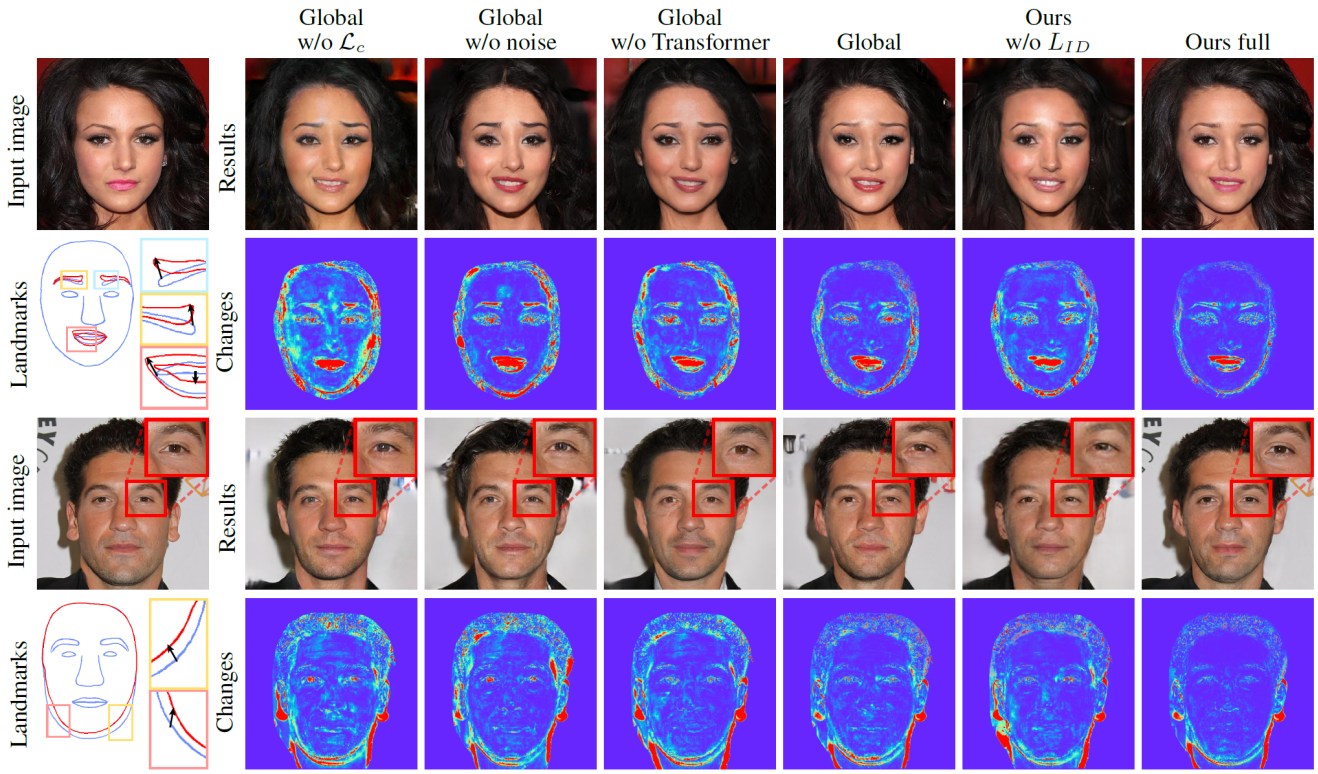


Figure 9: Ablation study. We show the generated results under different settings and their corresponding heat maps, which visualize the difference between the generated results and the input images. To focus on the face region, we set the background regions in the heat maps with the same color. Our full framework achieves the best results with the most accurate changes. In the other settings, there are more or less undesired changes, such as the change of skin color in "Global w/o $\mathcal{L}_c$".

Table 3: The statistics of the usability study. The scores range from 1 (the worst) to 5 (the best). We use the official checkpoint and interface of MaskGAN [24] for the usability study.

Aspects	MaskGAN (Drawing)	DeepFace Editing	Ours (Dragging)
Quality	3.64	**4.31**	4.27
Identity control	3.27	4.46	**4.55**
Expectation fitness	3.64	**4.57**	4.45
Usability	3.18	4.12	**4.55**

4.5. User Study

To evaluate the convenience of our dragging interface and expectation of reshaping results, we conduct a usability study and a perceptive evaluation study, respectively.

For the usability study, we compared our dragging interface and two drawing interfaces of MaskGAN [24] and DeepFaceEditing[5], since they provide well-designed open-source interfaces and are thus suitable for comparison. 10 users (7 male and 3 female) were invited to participate. According to their work experience in art, the users can be divided into three groups: 1 professional user, 3 middle users and 6 novice users. First they were asked to use the two compared systems by completing several tasks: enlarging the mouth and eyes, raising the eyebrows, and reshaping the nose and cheeks. Then the users were asked to give scores on the quality of results, identity control, expectation fitness, and usability. The results are listed in Tab. 3. In the aspect of quick editing, our system is considered to be easier to use than MaskGAN. For example, to open the mouth using their drawing system, users had to erase the original mask and then draw the upper lip, lower lip and inner mouth. In contrast, this is achieved with a simple dragging operation with our system.

For the perceptive evaluation study, we conducted an online questionnaire survey. 20 reshaping results by the participants and the authors with our tool were collected. The generation results by CocosNet, MaskGAN, SEAN, and ours were arranged in groups of four and the orders in groups were shuffled. Since MLS [36] and Bilayer [49] are not originally designed for face editing and their results vary little when the landmarks are not changed greatly, we

Table 4: The summarized results of the perceptive evaluation study. We report the average ranking score of the four compared methods (1: the best and 4: the worst). Our method scores better in all the aspects of quality, expectation fitness, and identity control compared to the other deep generative methods.

Aspects	MaskGAN	SEAN	CocosNet	Ours
Quality	3.01	2.88	2.12	**1.99**
Expectation fitness	2.43	2.59	2.25	**1.73**
Identity control	3.08	3.04	1.99	**1.89**

Table 5: Comparisons of computational cost with the existing methods. We measure the cost by multiply–accumulate operations (MACs). Our model has medium computational complexity.

Method	Bilayer	SEAN	CocosNet	MaskGAN	Ours
MACs(G)	17.3	343.2	380.2	18.3	48.2

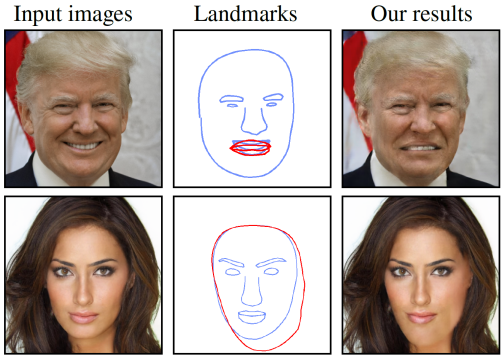


Figure 10: Limitations of our work. The input and output images in the first row show a failure case where the identity of the in-the-wild image is not preserved well: the areas of hair and skin are blurry. The input and output images in the second row reveal the limitation of deformation: the jaw has been reshaped out of normal range with the inner facial components unchanged, causing ugly generated results.

do not include the results by these techniques for fairness and to avoid confusing participants. The participants were asked to sort results from high to low in three aspects of generation quality, expectation fitness, and the ability to control identity. The results are summarized in Table 4.

4.6. Computational Cost

Since our framework divides the input image into four components and builds one partial network for each component, there is a potential concern that our method might require substantial computing power. To show the efficiency of our method, we compare the computational cost of our framework with the other deep learning methods. Here we do not compare the computation cost of MLS [36] with other methods calculated on the GPUs because MLS dose not involve extensive convolution operations and is mainly calculated on the CPU. As showed in Table 5, the computational cost does not increase a lot when we build one network for each component, since the total areas of inner components are far smaller than the whole image, the increased cost for the "mouth", "nose" and "eyebrows and eyes" is no more than the cost for the background. Meanwhile, the four parts can be generated in parallel and thus the computing time is basically up to the size of the whole image.

5. Conclusion and Discussions

We have proposed a novel deep generative framework for interactive face reshaping via an easy-to-use dragging interface. We apply neural shape deformation to the landmark graph so that users can provide precise control via a small number of landmark-based handles while our system automatically completes the desired deformation effects. Our local-to-global generative network produces realistic faces respecting user inputs and the identity of input face images. The effectiveness of our method and the usability of our interface have been confirmed by extensive experiments.

Although our model is able to synthesize edited facial images with high authenticity and details, there are still several limitations. First, the deformation is limited to the landmark graph domain, and thus our tool cannot be used to directly edit non-landmark regions like hair and accessories. A more general deformation domain for 2D facial images is needed and worthy of further study. Second, our framework might not cover extreme deformations that are outside the normal range, as shown in Fig. 10. This can be addressed by softening the constraints of handle points to restrict the user-manipulated landmarks to normal ranges. Third, the capability to preserve the identity of face images in the wild that are far away from the distribution of the training set is limited. For example as shown in the supplementary materials, the identity of the elderly is difficult to maintain, possibly because of the lack of old people in our training dataset. A possible solution is to enlarge the variety of training datasets and optimize the structure of network model. Besides the above limitations, since our tool does not support the direct control of details, our tool may have less granularity for professional editing compared to professional drawing editing tools such as DeepFaceEditing [5]. In the future, we are interested in extending neural deformation to the wider forms of representation beyond facial landmarks.

Acknowledgement

This work was supported by grants from the Open Research Projects of Zhejiang Lab(No. 2021KE0AB06), the National Natural Science Foundation of China (No. 62061136007 and No. 62102403), the Beijing Municipal Natural Science Foundation for Distinguished Young Scholars (No. JQ21013), the Open Project Program of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (No.VRLAB2022C07) and the Youth Innovation Promotion Association CAS.

References

[1] Face++. https://www.faceplusplus.com/dense-facial-landmarks/. 7

[2] Y. Alaluf, O. Patashnik, and D. Cohen-Or. Only a matter of style: Age transformation using a style-based regression model, 2021. 3

[3] H. Averbuch-Elor, D. Cohen-Or, J. Kopf, and M. F. Cohen. Bringing portraits to life. ACM Transactions on Graphics (TOG), 36(6):1–13, 2017. 3

[4] V. Blanz and T. 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

[5] S.-Y. Chen, F.-L. Liu, Y.-K. Lai, P. L. Rosin, C. Li, H. Fu, and L. Gao. DeepFaceEditing: Deep generation of face images from sketches. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2021), 40(4):90:1–90:15, 2021. 4, 10, 11

[6] S.-Y. Chen, W. Su, L. Gao, S. Xia, and H. Fu. Deepfacedrawing: deep generation of face images from sketches. ACM Transactions on Graphics (TOG), 39(4):72–1, 2020. 1, 2, 5

[7] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pages 4690–4699, 2019. 6, 7

[8] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis, 2020. 3, 5

[9] X. Gong, W. Chen, T. Chen, and Z. Wang. Sandwich batch normalization. arXiv preprint arXiv:2102.11382, 2021. 5

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014. 2

[11] S. Gu, J. Bao, H. Yang, D. Chen, F. Wen, and L. Yuan. Mask-guided portrait editing with conditional gans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3436–3445, 2019. 1, 3

[12] X. Han, C. Gao, and Y. Yu. Deepsketch2face: a deep learning based sketching system for 3d face and caricature modeling. ACM Transactions on graphics (TOG), 36(4):1–12, 2017. 1, 4

[13] X. Han, K. Hou, D. Du, Y. Qiu, S. Cui, K. Zhou, and Y. Yu. Caricatureshop: Personalized and photorealistic caricature sketching. IEEE transactions on visualization and computer graphics, 2018. 3, 4

[14] 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. In Advances in Neural Information Processing Systems, pages 6626–6637, 2017. 7

[15] A. Horé and D. Ziou. Is there a relationship between peak-signal-to-noise ratio and structural similarity index measure? IET Image Processing, 7(1):12–24, 2013. 8

[16] Y. Huang, Y. Wang, Y. Tai, X. Liu, P. Shen, S. Li, J. Li, and F. Huang. Curricularface: adaptive curriculum learning loss for deep face recognition. In proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 5901–5910, 2020. 7

[17] D. A. Hudson and C. L. Zitnick. Generative adversarial transformers. arXiv preprint, 2021. 3, 5

[18] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 2

[19] Y. Jiang, S. Chang, and Z. Wang. Transgan: Two transformers can make one strong gan. arXiv preprint arXiv:2102.07074, 2021. 3, 5

[20] Y. Jo and J. Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In Proceedings of the IEEE International Conference on Computer Vision, pages 1745–1753, 2019. 1, 3

[21] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4401–4410, 2019. 2, 5, 6, 7

[22] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of StyleGAN. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020. 2, 5

[23] P. Kaufmann, O. Wang, A. Sorkine-Hornung, O. Sorkine-Hornung, A. Smolic, and M. Gross. Finite element image warping. In Computer Graphics Forum, volume 32, pages 31–39. Wiley Online Library, 2013. 3

[24] 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, pages 5549–5558, 2020. 1, 2, 4, 5, 7, 8, 10

[25] T. Leyvand, D. Cohen-Or, G. Dror, and D. Lischinski. Data-driven enhancement of facial attractiveness. In ACM SIGGRAPH 2008 Papers, SIGGRAPH '08, New York, NY, USA, 2008. Association for Computing Machinery. 3

[26] H. Li, T. Weise, and M. Pauly. Example-based facial rigging. Acm transactions on graphics (tog), 29(4):1–6, 2010. 1

[27] O. Litany, A. Bronstein, M. Bronstein, and A. Makadia. Deformable shape completion with graph convolutional autoencoders. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1886–1895, 2018. 2, 5

[28] M. Mirza and S. Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014. 2

[29] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. Semantic image synthesis with spatially-adaptive normalization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2337–2346, 2019. 2

[30] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran. Image transformer. In International Conference on Machine Learning, pages 4055–4064. PMLR, 2018. 3

[31] S. Pidhorskyi, D. A. Adjeroh, and G. Doretto. Adversarial latent autoencoders. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2020. [to appear]. 3

[32] T. Portenier, Q. Hu, A. Szabo, S. A. Bigdeli, P. Favaro, and M. Zwicker. Faceshop: Deep sketch-based face image editing. arXiv preprint arXiv:1804.08972, 2018. 1, 3

[33] 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. arXiv preprint arXiv:2008.00951, 2020. 3

[34] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015. 6

[35] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Scribbler: Controlling deep image synthesis with sketch and color. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pages 5400–5409, 2017. 2

[36] S. Schaefer, T. McPhail, and J. Warren. Image deformation using moving least squares. In ACM SIGGRAPH 2006 Papers, pages 533–540. 2006. 5, 7, 8, 10, 11

[37] Y. Shen, J. Gu, X. Tang, and B. Zhou. Interpreting the latent space of gans for semantic face editing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020. 3

[38] Z. Tan, M. Chai, D. Chen, J. Liao, Q. Chu, L. Yuan, S. Tulyakov, and N. Yu. Michigan: multi-input-conditioned hair image generation for portrait editing. ACM Transactions on Graphics (TOG), 39(4):95–1, 2020. 5

[39] X. Tang, W. Sun, Y.-L. Yang, and X. Jin. Parametric reshaping of portraits in videos. In Proceedings of the 29th ACM International Conference on Multimedia, MM '21, page 4689–4697, New York, NY, USA, 2021. Association for Computing Machinery. 3

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017. 3

[41] T.-C. Wang, M.-Y. Liu, A. Tao, G. Liu, J. Kautz, and B. Catanzaro. Few-shot video-to-video synthesis. In Conference on Neural Information Processing Systems (NeurIPS), 2019. 2

[42] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro. Video-to-video synthesis. arXiv preprint arXiv:1808.06601, 2018. 2

[43] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 2, 6

[44] Z. Wang and A. C. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. IEEE signal processing magazine, 26(1):98–117, 2009. 8

[45] 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, 13(4):600–612, 2004. 7

[46] Q. Xiao, X. Tang, Y. Wu, L. Jin, Y.-L. Yang, and X. Jin. Deep shapely portraits. In Proceedings of the 28th ACM International Conference on Multimedia, MM '20, page 1800–1808, New York, NY, USA, 2020. Association for Computing Machinery. 3

[47] T. Xiao, J. Hong, and J. Ma. Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In Proceedings of the European conference on computer vision (ECCV), pages 168–184, 2018. 3

[48] S. Yang, Z. Wang, J. Liu, and Z. Guo. Deep plastic surgery: Robust and controllable image editing with human-drawn sketches. In European Conference on Computer Vision, 2020. 4

[49] E. Zakharov, A. Ivakhnenko, A. Shysheya, and V. Lempitsky. Fast bi-layer neural synthesis of one-shot realistic head avatars. In European Conference of Computer vision (ECCV), August 2020. 7, 8, 10

[50] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky. Few-shot aeversarial learning of realistic neural talking head models. In Proceedings of the IEEE International Conference on Computer Vision, pages 9459–9468, 2019. 5

[51] P. Zhang, B. Zhang, D. Chen, L. Yuan, and F. Wen. Cross-domain correspondence learning for exemplar-based image translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5143–5153, 2020. 7, 8

[52] L. Zhao, F. Han, X. Peng, X. Zhang, M. Kapadia, V. Pavlovic, and D. N. Metaxas. Cartoonish sketch-based face editing in videos using identity deformation transfer. Computers & Graphics, 79:58–68, 2019. 3

[53] S. Zhou, H. Fu, L. Liu, D. Cohen-Or, and X. Han. Parametric reshaping of human bodies in images. ACM transactions on graphics (TOG), 29(4):1–10, 2010. 2

[54] P. Zhu, R. Abdal, Y. Qin, and P. Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020. 1, 2, 3, 6, 7, 8