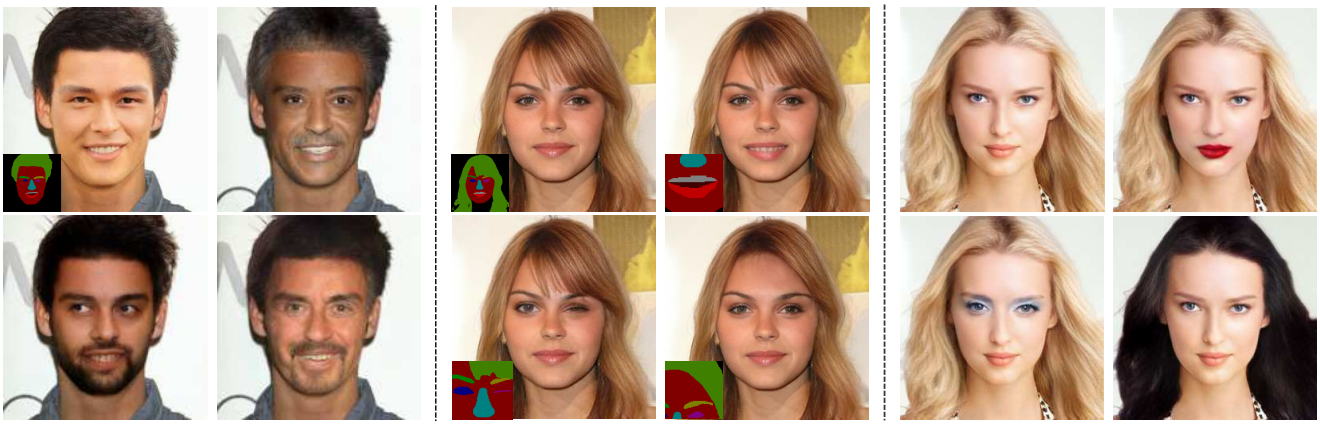
Mask-Guided Portrait Editing with Conditional GANs

Shuyang Gu[1] Jianmin Bao[1] Hao Yang[2] Dong Chen[2] Fang Wen[2] Lu Yuan[2]
[1]University of Science and Technology of China [2]Microsoft Research
{gsy777,jmbao}@mail.ustc.edu.cn {haya,doch,fangwen,luyuan}@microsoft.com



(a) Mask2image (b) Component editing (c) Component transfer

Figure 1: We propose a framework based on conditional GANs for mask-guided portrait editing. (a) Our framework can generate diverse and realistic faces using one input target mask (lower left corner in the first image). (b) Our framework allows us to edit the mask to change the shape of face components, *i.e.* mouth, eyes, hair. (c) Our framework also allows us to transfer the appearance of each component for a portrait, including hair color.

Abstract

*Portrait editing is a popular subject in photo manipulation. The Generative Adversarial Network (GAN) advances the generating of realistic faces and allows more face editing. In this paper, we argue about three issues in existing techniques: diversity, quality, and controllability for portrait synthesis and editing. To address these issues, we propose a novel end-to-end learning framework that leverages conditional GANs guided by provided face masks for generating faces. The framework learns feature embeddings for every face component (*e.g.*, mouth, hair, eye), separately, contributing to better correspondences for image translation, and local face editing. With the mask, our network is available to many applications, like face synthesis driven by mask, face Swap+ (including hair in swapping), and local manipulation. It can also boost the performance of face parsing a bit as an option of data augmentation.*

1. Introduction

Portrait editing is of great interest in the vision and graphics community due to its potential applications in movies, gaming, photo manipulation and sharing, etc. People enjoy the magic that makes faces look more interesting, funny, and beautiful, which appear in an amount of popular apps, such as Snapchat, Facetune, etc.

Recently, advances in Generative Adversarial Networks (GANs) [16] have made tremendous progress in synthesizing realistic faces [1, 29, 25, 12], like face aging [46], pose changing [44, 21] and attribute modifying [4]. However, these existing approaches still suffer from some quality issues, like lack of fine details in skin, difficulty in dealing with hair and background blurring. Such artifacts cause generated faces to look unrealistic.

To address these issues, one possible solution is to use the facial mask to guide generation. On one hand, a face mask provides a good geometric constraint, which helps synthesize realistic faces. On the other hand, an accurate contour for each facial component (*e.g.*, eye, mouth, hair, etc.) is necessary for local editing. Based on the face mask, some works [40, 14] achieve very promising results in portrait stylization. However, these methods focus on transferring the visual style (*e.g.*, B&W, color, painting) from the reference face to the target face. It seems to be unavailable for synthesizing different faces, or changing face components.

Some kinds of GAN models begin to integrate the face mask/skeleton for better image-to-image translation, for

example, pix2pix [24], pix2pixHD [45], where the facial skeleton plays an important role in producing realistic faces and enabling further editing. However, the diversity of their synthesized faces are so limited, for example, the input and output pairs might not allow noticeable changes in emotion. The quality issue especially on hair and background prevents resulting images from being realistic. The recent work BicycleGAN [48] tries to generate diverse faces from one input mask, but the diversity is limited on color or illumination.

We have to reinvestigate and carefully design the image-to-image translation model, which addresses the three problems – *diversity*, *quality*, and *controllability*. Diversity requires the learning of good correspondences between image pairs, which may undergo variance in poses, lightings, colors, ages and genders, for image translation. Quality should further improve in fine facial details, hair, and background. More controls for local facial components are also key.

In this paper, we propose a framework based on conditional GANs [33] for portrait editing guided by face masks. The framework consists of three major components: *local embedding sub-network*, *mask-guided generative sub-network*, and *background fusion sub-network*. The three sub-networks are trained end-to-end. The *local embedding sub-network* involves five auto-encoder networks which respectively encode embedding information for five facial components, *i.e*., "left eye", "right eye", "mouth", "skin & nose", and "hair". The face mask is used to help specify the region for each component in learning. The *mask-guided generative sub-network* recombines the pieces of local embeddings and the target face mask together, yielding the foreground face image. The face mask helps establish correspondences at the component level (*e.g*., mouth-to-mouth, hair-to-hair, etc.) between the source and target images. At the end, the *background fusing sub-network* fuses the background and the foreground face to generate a natural face image, according to the target face mask. For guidance, the face mask aids facial generation in all of three sub-networks.

With the mask, our framework allows many applications. As shown in Figure 1 (a), we can generate new faces driven by the face mask, *i.e*., mask-to-face, as well as skeleton-to-face in [24, 45]. We also allow more editing, such as removing hairs, amplifying or reducing eyes, and making it smile, as shown in Figure 1 (b). Moreover, we can modify the appearance of existing faces locally, such as the changing appearance of each facial component, shown in Figure 1 (c). Experiments shows that our methods outperform state-of-the-art face synthesis driven by a mask (or skeleton) in terms of diversity and quality. More interesting, our framework can help boost the performance of face parsing algorithm marginally as the data augmentation.

Overall, our contributions are as follows:

1. We propose a novel framework based on mask-guided conditional GANs, which successfully addresses diversity, quality and controllability issues in face synthesis.
2. The framework is general and available for an amount of applications, such as mask-to-face synthesis, face editing, face swap+, and even data augmentation for face parsing.

2. Related Work

Generative Adversarial Networks Generative adversarial networks (GANs) [16] have achieved impressive results in many directions. It forces the generated samples to be indistinguishable from the target distribution by introducing an adversarial discriminator. The GAN family enables a wide variety of computer vision applications such as image synthesis [2, 36], image translation [24, 47, 45, 48], and representation disentangling [22, 4, 44], among others.

Inspired by the conditional GAN models [33] that generate images from masks, we propose a novel framework for mask-guided portrait editing. Our method leverages local embedding information for individual facial components, generating portrait images with higher diversity and controllability, than existing global-based approaches such as pix2pix [24] or pix2pixHD [45].

Deep Visual Manipulation Image editing has benefited a lot from the rapid growth of deep neural networks, including image completion [23], super-resolution [28], deep analogy [30], and sketch-based portrait editing [39, 34], to name a few. Among them, the most related are mask-guided image editing methods, which train deep neural networks to translate masks into realistic images [11, 8, 24, 45, 47].

Our approach also relates to the visual attribute transfer methods, including style transfer [15, 17] and color transfer [19]. Recently, the Paired-CycleGAN [9] has been proposed for makeup transfer, in which a makeup transfer function and a makeup removal function are trained in pair. Though similar, the appearances of facial instances that our method disentangles differ from makeups. For example, the color and curly types of hairs which we can transfer are definitely not makeups. Furthermore, there are some works focusing on editing a specific component in faces (*e.g*., eyes [13, 41]) or editing attributes of faces [35, 42].

With the proposed structure that disentangles and recombines facial instance embeddings with face masks, our method also enhances over face swapping methods [5, 26] by supporting explicit face and hair swapping.

Non-Parametric Visual Manipulation Non-parametric image synthesis approaches [18, 6, 27] usually generate new images by warping and stitching together existing patches from a database. The idea is extended by Qi *et*

al. [37] which combines neural networks to improve quality. Though similar at first glance, our method is intrinsically different from non-parametric image synthesis: our local embedding sub-network encodes facial instances as embeddings instead of image patches. New face images in our method are generated through the mask-guided generative sub-network, instead of warping and stitching image patches together. By jointly training all sub-networks, our model generates facial images that are higher quality than non-parametric methods that may also be difficult for them: *e.g*. synthesizing a face with an open mouth showing teeth from a source face with a closed mouth hiding all teeth.

3. Mask-Guided Portrait Editing Framework

We propose a framework based on conditional GANs for mask-guided portrait editing. Our framework requires four inputs, a source image $\boldsymbol{x^s}$, the mask of source image $\boldsymbol{m^s}$, a target image $\boldsymbol{x^t}$, and the mask of target image $\boldsymbol{m^t}$. The mask $\boldsymbol{m^s}$ and $\boldsymbol{m^t}$ can be obtained by a face parsing network. If we want to change the mask, we can manually edit $\boldsymbol{m^t}$. With the source mask $\boldsymbol{x^s}$, we cannot get the appearance of each face component, *e.g*., "left eye", "right eye", "mouth", "skin & nose", and "hair" from the source image. With the target mask $\boldsymbol{m^t}$, we can get the background from the target image $\boldsymbol{x^t}$. Our framework first recombines the appearance of each component from $\boldsymbol{x^s}$ and the target mask together, yielding the foreground face, then fusing it with the background from $\boldsymbol{x^t}$, outputting the final result $G(\boldsymbol{x^s}, \boldsymbol{m^s}, \boldsymbol{x^t}, \boldsymbol{m^t})$. G indicates the overall generative framework.

As shown in Figure 2, we first use a *local embedding sub-network* to learn feature embedding for the input source image $\boldsymbol{x^s}$. It involves five auto-encoder networks to encode embedding information for five facial components respectively. Comparing with [45, 48] which learn global embedding information, our approach can retain more source facial details. The *mask guided generative sub-network* then specifies the region of each embedded component feature and concatenates all features of the five local components together with the target mask to generate the foreground face. Finally, we use the *background fusing sub-network* to fuse the foreground face and the background to generate a natural facial image.

3.1. Framework Architecture

Local Embedding Sub-Network. To enable component-level controllability of our framework, we propose learning feature embeddings for each component in the face. We first use a face parsing network P_F (details in Section 3.2) which is a Fully Convolution Network (FCN) trained on the Helen dataset [43] to get the source mask $\boldsymbol{m^s}$ of the source image $\boldsymbol{x^s}$. Then, according to the face mask, we segment the foreground face image into five components $\boldsymbol{x_i^s}$, $i \in \{0, 1, 2, 3, 4\}$, *i.e*., "left eye", "right eye", "mouth", "skin & nose", and "hair". For each facial component, we use the corresponding auto-encoder network $\{E^i_{local}, G^i_{local}\}$, $i \in \{0, 1, 2, 3, 4\}$, to embed its component information. With five auto-encoder networks, we can conveniently change any one of facial components in the generated face images or recombine different components from different faces.

Previous works, *e.g*., pix2pixHD [45], also train an auto-encoder network to get the feature vector that corresponds with each instance in the image. To guarantee the features that fit different instance shape, they add an instance-wise average pooling layer to the output of the encoder to compute the average feature for the object instance. Although this approach allows object-level control on the generated results, their generated faces still suffer from low quality for two reasons. First, they use a global encoder network to learn feature embeddings for different instances in the image. We argue that merely a global network is quite limited in learning and recovering all local details of each instance. Second, the instance-wise average pooling would remove many characteristic details in reconstruction.

Mask-Guided Generative Sub-Network. To make the target mask $\boldsymbol{m^t}$ a guidance for mask equivariant facial generation, we adopt an intuitive way to fuse the five component feature tensors and the mask feature tensor together. As shown in Figure 2 (b), five component feature tensors are extracted by the *local embedding sub-network*, and the mask feature tensor is the output of the encoder E_m.

First, we get the center location $\{c_i\}_{i=1...5}$ of each component from the target mask $\boldsymbol{x^t}$. Then we prepare five 3D tensors all filled with 0, *i.e*., $\{\hat{\boldsymbol{f}}_i\}_{i=1...5}$. Every tensor has the same height and width with the mask feature tensor, and the same channel number with each component feature tensor. Next, we copy each of the five learned component feature tensors to all-zero tensor $\hat{\boldsymbol{f}}_i$ centered at c_i according to the target mask (*e.g*., mouth-to-mouth, eye-to-eye etc.). After that, we concatenate all 3D and mask feature tensors to produce a fused feature tensor. Finally, we feed the fused feature tensor to the network G_m and produce the foreground face image.

Background Fusing Sub-Network. To paste the generated foreground faces to the background of the target image, the straightforward approach is to copy the background from the target image $\boldsymbol{x^t}$ and combine it with the foreground faces according to the target face mask. However, this causes noticeable boundary artifacts in the final results. There are two possible reasons. First, the background contains neck skin parts, so the unmatched face skin color in the source face $\boldsymbol{x^s}$ and the neck skin color in the target image $\boldsymbol{x^t}$ cause the artifacts. Second, the segmentation mask for the hair part is not always perfect, so the hair in the back-

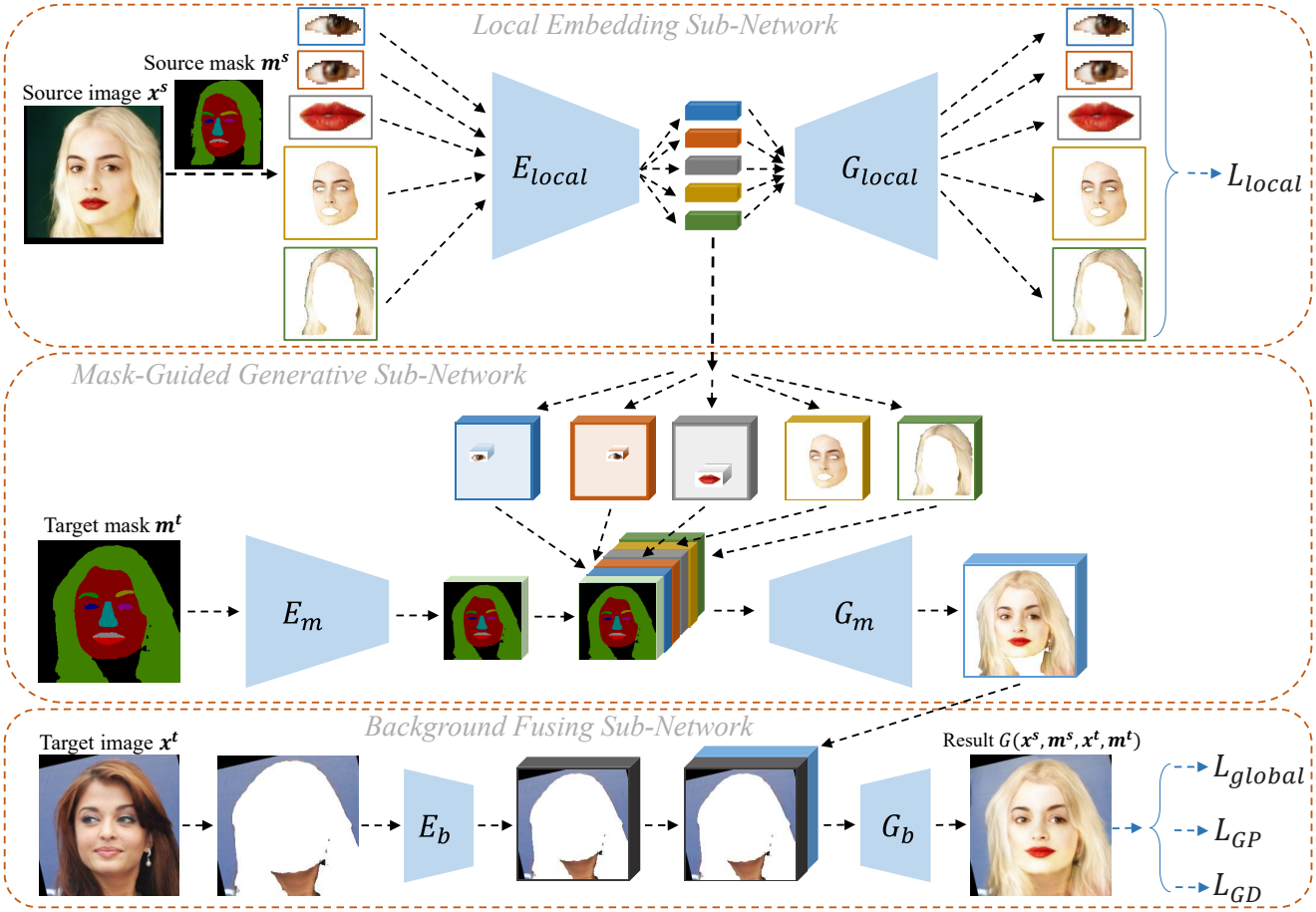


Figure 2: The proposed framework for mask-guided portrait editing. It contains three parts: *local embedding sub-network*, *mask guided generative sub-network*, and *background fusing sub-network*. *Local embedding sub-network* learns the feature embedding of the local components of the source image. *Mask guided sub-network* combines the learned component feature embeddings and mask to generate the foreground face image. *Background fusing sub-network* generates the final result from the foreground face and the background. The loss functions are drawn with the blue dashed lines.

ground also causes artifacts.

To solve this problem, we propose using the *background fusing sub-network* to remove the artifacts in fusion. We first use the face parsing network P_F (details in Section 3.2) to get the target face mask $\boldsymbol{x^t}$. According to the face mask, we extract the background part from the target image, and then feed the background part to an encoder E_b to obtain the output background feature tensor. After that, we concatenate the background feature tensor with the foreground face, and feed it to the generative network G_b producing facial result $G(\boldsymbol{x^s}, \boldsymbol{m^s}, \boldsymbol{x^t}, \boldsymbol{m^t})$.

3.2. Loss Functions

Local Reconstruction. We use the MSE loss between the input instances and the reconstructed instances to learn the feature embedding of each instance.

$$\mathcal{L}_{local} = \frac{1}{2}||\boldsymbol{x_i^s} - G^i_{local}(E^i_{local}(\boldsymbol{x_i^s}))||_2^2, \qquad (1)$$

where $\boldsymbol{x_i^s}$, $i \in \{0, 1, 2, 3, 4\}$ represents "left eye", "right eye", "mouth", "skin & nose", and "hair" in $\boldsymbol{x^s}$.

Global Reconstruction. We consider the reconstruction error in training. When the input source images $\boldsymbol{x^s}$ is the same as the target image $\boldsymbol{x^t}$, the generated result $G(\boldsymbol{x^s}, \boldsymbol{x^s}, \boldsymbol{x^t}, \boldsymbol{m^t})$ should be the same as $\boldsymbol{x^s}$. Based on the constraint, the reconstruction loss can be measured by:

$$\mathcal{L}_{global} = \frac{1}{2}||G(\boldsymbol{x^s}, \boldsymbol{m^s}, \boldsymbol{x^t}, \boldsymbol{m^t}) - \boldsymbol{x^s}||_2^2 \qquad (2)$$

Adversarial Loss. To produce realistic results, we add discriminator networks D after the framework. Similar to GAN, the overall framework G plays a minimax game with discriminator network D. Since a simple discriminator network D is not suitable for face image synthesis with resolution 256×256. Following the method in pix2pixHD [45], we also use multi-scale discriminators. We use 2 discriminators that have an identity network structure but operate at different image scales. we downsample the real and generated samples by a factor of 2 using the average pooling layer. Moreover, the generated samples should be conditioned on the target mask $\boldsymbol{m^t}$. So the loss function for the discriminators $D_i, i \in 1, 2$ is:

$$\mathcal{L}_{D_i} = -\mathbb{E}_{\boldsymbol{x}^t \sim P_r}[\log D_i(\boldsymbol{x}^t, \boldsymbol{m}^t)] - \mathbb{E}_{\boldsymbol{x}^t, \boldsymbol{x}^s \sim P_r}[\log(1 - D_i(G(\boldsymbol{x}^s, \boldsymbol{m}^s, \boldsymbol{x}^t, \boldsymbol{m}^t), \boldsymbol{m}^t)], \tag{3}$$

and the loss function for the framework G is:

$$\mathcal{L}_{sigmoid} = -\mathbb{E}_{\boldsymbol{x}^t, \boldsymbol{x}^s \sim P_r}[\log(D_i(G(\boldsymbol{x}^s, \boldsymbol{m}^s, \boldsymbol{x}^t, \boldsymbol{m}^t), \boldsymbol{m}^t)]. \tag{4}$$

The original loss function for G may cause unstable gradient problems. Inspired by [3, 45], we also use a pairwise feature matching objective for the generator. To generate realistic face images quality, we match the features of the network D between real and fake images. Let $\boldsymbol{f}_{\boldsymbol{D}_i}(\boldsymbol{x}, \boldsymbol{m}^t)$ denote features on an intermediate layer of the discriminator, then the pairwise feature matching loss is the Euclidean distance between the feature representations, *i.e.*,

$$\mathcal{L}_{FM} = \frac{1}{2}||\boldsymbol{f}_{\boldsymbol{D}_i}(G(\boldsymbol{x}^s, \boldsymbol{m}^s, \boldsymbol{x}^t, \boldsymbol{m}^t), \boldsymbol{m}^t) - \boldsymbol{f}_{\boldsymbol{D}_i}(\boldsymbol{x}^t, \boldsymbol{m}^t)||_2^2, \tag{5}$$

where we use the last output layer of network D_i as the feature $\boldsymbol{f}_{\boldsymbol{D}_i}$ for our experiments.

The overall loss from the discriminator networks to the framework G is:

$$\mathcal{L}_{GD} = \mathcal{L}_{sigmoid} + \lambda_{FM}\mathcal{L}_{FM}. \tag{6}$$

where λ_{FM} controls the importance of the two terms.

Face Parsing Loss. In order to generate mask equivariant facial images, we need to make the generated samples have the same mask as the target mask, and a face paring network P_F to constrain the generated faces, following previous methods [32, 38]. We pretrain the face parsing network P_F with a U-Net network structure on the Helen Face Dataset [43]. The loss function $\mathcal{L}_P$ for network P_F is the pixel-wise cross entropy loss, This loss examines each pixel individually, comparing the class predictions (depth-wise pixel vector) to our one-hot encoded target vector $p_{i,j}$:

$$\mathcal{L}_P = -\mathbb{E}_{\boldsymbol{x} \sim P_r}[\sum_{i,j} \log P(p_{i,j}|P_F(\boldsymbol{x})_{i,j})]. \tag{7}$$

Here, the (i, j) indicates the location of the pixel.

After we get the pretrained network P_F. we use P_F to encourage the generated samples to have the same mask with the target mask, so we use the following loss function for the generative network:

$$\mathcal{L}_{GP} = -\mathbb{E}_{\boldsymbol{x} \sim P_r}[\sum_{i,j} \log P(\boldsymbol{m}^t_{i,j}|P_F(G(\boldsymbol{x}^s, \boldsymbol{m}^s, \boldsymbol{x}^t, \boldsymbol{m}^t))_{i,j})], \tag{8}$$

where $\boldsymbol{m}^t_{i,j}$ is the ground truth label of $\boldsymbol{x}^t$ located at (i, j). $P_F(G(\boldsymbol{x}^s, \boldsymbol{m}^s, \boldsymbol{x}^t, \boldsymbol{m}^t))_{i,j}$ is the predict pixel located at (i, j).

Overall Loss Functions. The final loss for G is a sum of the above losses in Equation 1, 2, 6, 8.

$$\mathcal{L}_G = \lambda_{local}\mathcal{L}_{local} + \lambda_{global}\mathcal{L}_{global} + \lambda_{GD}\mathcal{L}_{GD} + \lambda_{GP}\mathcal{L}_{GP}, \tag{9}$$

where λ_{local}, λ_{global}, λ_{GD}, and λ_{GP} are the trade-offs balancing different losses. In our experiments, λ_{local}, λ_{global}, λ_{GD}, and λ_{GP} are set as $\{10, 1, 1, 1\}$ respectively.

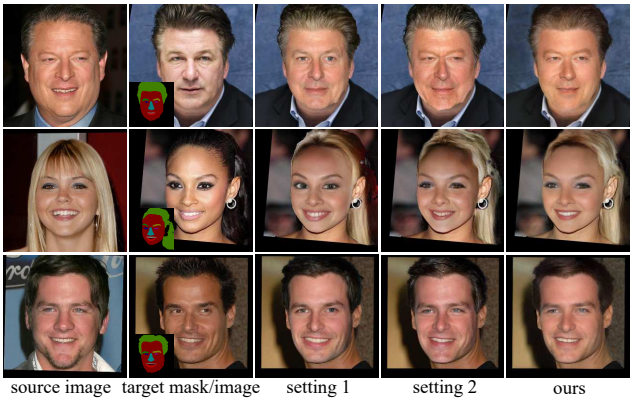


Figure 3: Visual comparison of our proposed framework and its variants.

3.3. Training Strategy

During training, the input masks $\boldsymbol{m}^s$, $\boldsymbol{m}^t$ always use the parsing results of source image $\boldsymbol{x}^s$ and target image $\boldsymbol{x}^t$. We consider two situations in training: 1) $\boldsymbol{x}^s$ and $\boldsymbol{x}^t$ are the same, which is called paired data, 2) $\boldsymbol{x}^s$ and $\boldsymbol{x}^t$ are different, which is called unpaired data. Inspired by [4] , we incorporate these settings into the training stage, and employ a $(1 + 1)$ strategy, one step for paired data training and the other step for unpaired data training. However, the training loss functions for these two settings should be different. For paired data, we use all losses in $\mathcal{L}_G$, but for unpaired data, we set λ_{global} and λ_{FM} to zero in $\mathcal{L}_G$.

4. Experiments

In this section, we validate the effectiveness of the proposed method. We evaluate our model on the Helen Dataset[43]. The Helen Dataset contains $2,330$ face images ($2,000$ for training and 330 for testing) with the pixel-level mask label annotated. But $2,000$ facial images have limited diversity, so we first use these $2,000$ face images to train a face parsing network, and use the parsing network to get semantic masks for an additional $20,000$ face images from VGGFace2 [7]. We use a total of $22,000$ face images for training during experiments . For all training faces, we first detect the facial region with the JDA face detector [10], and then locate five facial landmarks (two eyes, nose tip and two mouth corners). After that, we use similarity transformation based on the facial landmarks to align faces to a canonical position. Finally, we crop a 256×256 facial region to do the experiments.

In our experiments, the input size of five instances (left eye, right eye, mouth, skin, and hair) are decided by the max size of each component. Specially, we use 48×32, 48×32, 144×80, 256×256, 256×256 for left eye, right eye, mouth, skin, and hair in our experiments. For network details of E_{local}, G_{local}, E_m, G_m, E_b, and G_b and training settings, please refer to the supplementary material.

Figure 4: Comparison of mask to face synthesis results with pix2pixHD [45] and BicycleGAN[48], The target mask and the target face image are on the left side. The first and second rows are the generated results from pix2pixHD and BicyleGAN respectively. The diversity of the generated samples are mainly from the skin color or illumination. The third row is the generated results from our methods. We can generate more realistic and diverse facial images.

4.1. Analysis of the Proposed Framework

Our framework is designed to solve three problems – diversity, quality, and controllability in mask-guided facial synthesis. To validate this, we perform a step by step ablation study to understand how our proposed framework helps solve these three problems.

We perform three gradually changed settings to validate our framework: 1) We train a framework using a global auto-encoder to learn the global embedding of the source image, then we concatenate the global embedding with the target mask to generate the foreground face image with losses $\mathcal{L}_{global}$, $\mathcal{L}_{GP}$, and $\mathcal{L}_{GD}$. We then crop the background from the target image and directly paste it to the generated foreground face to get the result. 2) We train another framework using a *local embedding sub-network* to learn the embedding of each component of the source image, then we concatenate the local embedding with the target mask to generate the foreground face. After that, we get the background using the same method as 1). 3) We train our framework taking full advantage of *local embedding sub-network*, *mask-guided generative sub-network*, and *background fusing sub-network* to get the final results.

Figure 3 presents qualitative results for the above three settings. Comparing settings 2 and 1, we see that using a *local embedding sub-network* helps the generated results to keep the details (*e.g.* eye's size, skin color, hair color) from the source images. This enables the controllability of our framework to control each component of the generated face. By feeding different components to the *local embedding sub-network*, we can generate diverse results, which shows our framework handles the diversity problem. Comparing these two variant settings with our method, a background copied directly from the target image causes noticeable boundary artifacts. In our framework, the *background fusing sub-network* helps to remove the artifacts and generate more realistic faces, proving that our framework can generate high quality faces.

Table 1: Quantitative comparison of our framework and its variants, setting 1,2 are defined in Section 4.1.

Setting	1	2	ours
FID	11.02	11.26	8.92

Table 2: Comparison of face parsing results with and without using face parsing networks.

Method	avg. per-pixel accuracy
w/o face parsing networks	0.946
w face parsing networks	0.977

To quantitatively evaluate each setting, we generate $5,000$ facial images for each setting, and calculate the FID [20] between the generated faces and the training faces. In Table 1, we report the FID for each setting. The *local embedding sub-network* and *background fusing sub-network* help improve the quality of generated samples. Meanwhile, the low FID score indicates that our model can generate high-quality faces from masks.

To validate whether our face parsing loss helps keep the mask of the generated samples, we conduct an experiment to validate this. We train another framework without using the face parsing loss. Then we generate $5,000$ samples from this framework. Next, we use another set of face parsing networks to get the average per-pixel accuracy with the target mask as ground truth for all generated faces. Table 2 reports the results, showing that the face parsing loss helps to preserve the mask of the generated faces.

4.2. Mask-to-Face Synthesis

This section presents the results of mask to face synthesis. The goal of mask to face synthesis is to generate realistic, diverse and mask equivariant facial images from a given target mask. To demonstrate that our framework has the ability to generate realistic and diverse faces from an input mask, we choose some masks and randomly choose some facial images as the source image and synthesize the facial images.

Figure 5 presents the face synthesis results from the input masks. The first column is the target masks, and the face images on the right side are the generated face images conditioned on the target masks. We observe that the generated face images are photo-realistic. Meanwhile, the generated facial images perform well in terms of diversity, such as in skin color, hair color, eye makeup, and even beard. Furthermore, the generated facial images also maintain the mask.

Previous methods also try to do the mask-to-face synthesis, BicycleGAN [48] is an image-to-image translation model which can generate continuous and multimodal out-

Figure 5: Our framework can synthesize realistic, diverse and mask equivariant faces from one target mask.

put distributions. pix2pixHD allows high resolution image-to-image translation. In Figure 4, we show the qualitative comparison results. We observe that the generated samples by BicycleGAN and pix2pixHD show a limited diversity, and the diversity lies in the skin color or illumination. The reason for this is that they use a global encoder, so the generated samples cannot leverage the diversity of components in the faces. In contrast, the generated results by our methods look realistic, clear and diverse. Our model is also able to keep the mask information. It shows the strength of the proposed framework.

4.3. Face Editing and Face Swap+

Another important application for our framework is facial editing. With the changeable target mask and source image, we can edit on the generated face images by editing the target mask or replacing the facial component from the source image. We conduct two experiments: 1) changing the target mask to change the appearance of the generated faces; 2) replacing the facial component of the target image with new component from other's faces to explicitly change the corresponding component of the generated faces.

Figure 6 shows the generated results of changing the target mask. We replace the label of hair region on forehead with skin label, and we get a vivid face images with no hair on the forehead. Besides that, we can change the shape of the mouth, eyes, eyebrows region in the mask, and get an output facial image even with new emotions. This shows the effectiveness of our model in generating mask equivariant and realistic results.

Figure 7 shows the results of changing individual parts of the generated face. We can add the beards to the generated face by replacing the target skin part with the skin part of a face with a beard. We also change the mouth color, hair color, and even the eye makeup by changing the local embedding part. This shows the effectiveness of our model in generating local instance equivariant and realistic results.

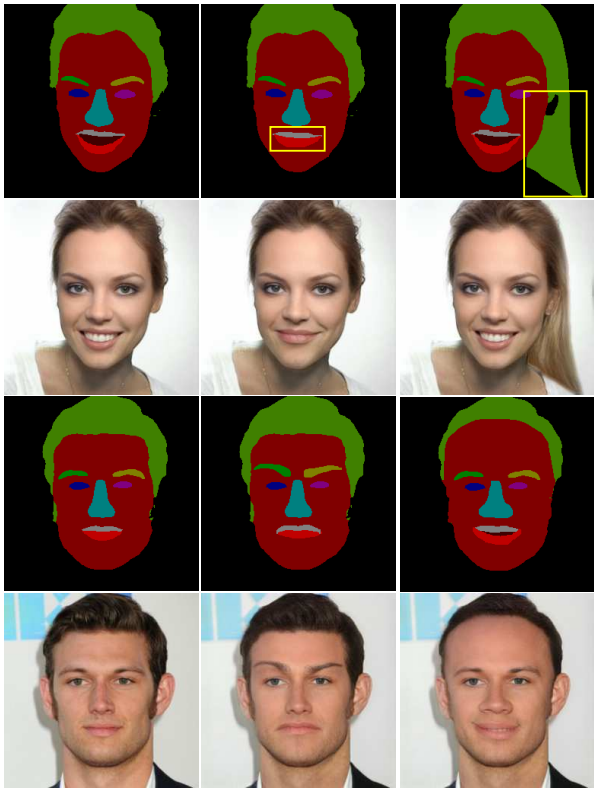


Figure 6: Our framework allows users to change mask labels locally to explicitly manipulate facial components, like making short hair become longer, and even changing the emotion.

Figure 8 shows the results of face swap+. Different from traditional face swap algorithm, our framework can not only swap the face appearance but also keep the component shape. More importantly, we can explicitly swap the hair part compared to previous methods.

Furthermore, Figure 9 shows more results for input face under extreme conditions. In the first two rows, the input

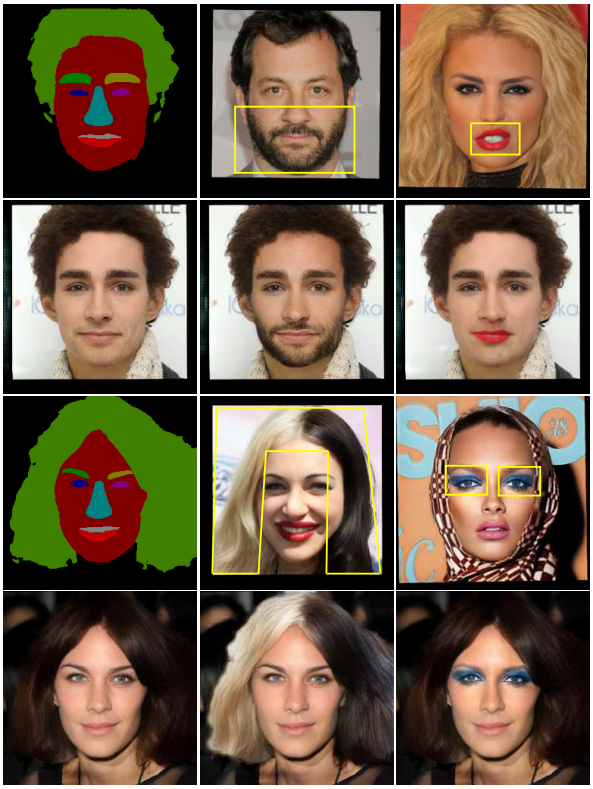


Figure 7: We can also edit the appearance of instances in the face, *e.g.* adding beards and changing the color of mouth, hair, and eye makeup.

Figure 8: Our framework can enhance an existing facial swap, called face swap+, to explicitly swap the hair.

face images have large poses and extreme illumination. Our method gets reasonable and realistic results. Also, if the input face has eye-glasses, we find that the result relies on the segmentation mask. If the glasses are labeled as background, it can be reconstructed by our *background fusion sub-network*, the generated result is shown in the last row.

4.4. Synthesized Faces for Face Parsing

We further show that the facial images synthesized from our framework can benefit the training the face parsing model. We use the $2,000$ masks of the trainings images of Helen Face Dataset as the target mask, and randomly choose face images from CelebA Dataset [31] as the source image to generate face images, then we remove these facial images generated with different genders. Then we conduct three experiments: 1) we only use the training set of the Helen Face Dataset to train a face parsing network without data augmentation; 2) we use the training set of the Helen Face Dataset to train a face parsing network with a data augmentation strategy (flip, geometry transform, scale, rotation); 3) we add the generated facial images to the Helen Face Dataset training set, using target mask as ground-truth and the same data augmentation strategy.

Figure 9: The generated results with input faces under extreme conditions (large pose, extreme illumination and face with glasses).

Table 3: Results of face parsing with added generated facial images.

Helen	0.728
Helen (with data augmentation)	0.863
Helen + generated (with data augmentation)	0.871

We use 100 test images from the Helen Face Dataset for testing. Table 3 shows the face parsing accuracy in three different settings. With the new generated faces, we get a 0.8% improvement in accuracy compared with no generated face images. This demonstrates that our generative framework has a certain extrapolation ability.

5. Conclusion

In this paper, we propose a novel end-to-end framework based on mask-guided conditional GANs, which can synthesize diverse, high-quality, and controllable facial images from given masks. With the changeable input facial mask and source image, our framework allows users to do high-level portrait editing, such as: explicitly editing face components in the generated face and transferring local appearances from one face to the generated face. We can even get a better face paring model by leveraging the synthesized facial data from input masks. Our experiments demonstrate the excellent performance of our framework.

References

[1] G. Antipov, M. Baccouche, and J.-L. Dugelay. Face aging with conditional generative adversarial networks. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 2089–2093. IEEE, 2017.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[3] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua. Cvae-gan: Fine-grained image generation through asymmetric training. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2764–2773. IEEE, 2017.

[4] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua. Towards open-set identity preserving face synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6713–6722, 2018.

[5] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, and S. K. Nayar. Face swapping: automatically replacing faces in photographs. In *ACM Transactions on Graphics (TOG)*, volume 27, page 39. ACM, 2008.

[6] P. P. Busto, C. Eisenacher, S. Lefebvre, M. Stamminger, et al. Instant texture synthesis by numbers. In *VMV*, pages 81–85, 2010.

[7] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face and Gesture Recognition*, 2018.

[8] A. J. Champandard. Semantic style transfer and turning two-bit doodles into fine artworks. *arXiv preprint arXiv:1603.01768*, 2016.

[9] H. Chang, J. Lu, F. Yu, and A. Finkelstein. PairedCycleGAN: Asymmetric style transfer for applying and removing makeup. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[10] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In *European Conference on Computer Vision*, pages 109–122. Springer, 2014.

[11] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *IEEE International Conference on Computer Vision (ICCV)*, volume 1, page 3, 2017.

[12] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *arXiv preprint*, 1711, 2017.

[13] B. Dolhansky and C. Canton Ferrer. Eye in-painting with exemplar generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7902–7911, 2018.

[14] J. Fišer, O. Jamriška, D. Simons, E. Shechtman, J. Lu, P. Asente, M. Lukáč, and D. Sỳkora. Example-based synthesis of stylized facial animations. *ACM Transactions on Graphics (TOG)*, 36(4):155, 2017.

[15] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423. IEEE, 2016.

[16] 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.

[17] S. Gu, C. Chen, J. Liao, and L. Yuan. Arbitrary style transfer with deep feature reshuffle. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8222–8231, 2018.

[18] J. Hays and A. A. Efros. Scene completion using millions of photographs. In *ACM Transactions on Graphics (TOG)*, volume 26, page 4. ACM, 2007.

[19] M. He, J. Liao, L. Yuan, and P. V. Sander. Neural color transfer between images. *arXiv preprint arXiv:1710.00756*, 2017.

[20] 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.

[21] R. Huang, S. Zhang, T. Li, R. He, et al. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. *arXiv preprint arXiv:1704.04086*, 2017.

[22] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. *arXiv preprint arXiv:1804.04732*, 2018.

[23] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017.

[24] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

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

[26] I. Korshunova, W. Shi, J. Dambre, and L. Theis. Fast face-swap using convolutional neural networks. In *The IEEE International Conference on Computer Vision*, pages 3697–3705, 2017.

[27] J.-F. Lalonde, D. Hoiem, A. A. Efros, C. Rother, J. Winn, and A. Criminisi. Photo clip art. *ACM transactions on graphics (TOG)*, 26(3):3, 2007.

[28] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, volume 2, page 4, 2017.

[29] M. Li, W. Zuo, and D. Zhang. Convolutional network for attribute-driven and identity-preserving human face generation. *arXiv preprint arXiv:1608.06434*, 2016.

[30] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang. Visual attribute transfer through deep image analogy. *arXiv preprint arXiv:1705.01088*, 2017.

[31] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[32] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

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

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

[35] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 818–833, 2018.

[36] G.-J. Qi. Loss-sensitive generative adversarial networks on lipschitz densities. *arXiv preprint arXiv:1701.06264*, 2017.

[37] X. Qi, Q. Chen, J. Jia, and V. Koltun. Semi-parametric image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8808–8816, 2018.

[38] 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.

[39] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

[40] Y. Shih, S. Paris, C. Barnes, W. T. Freeman, and F. Durand. Style transfer for headshot portraits. *ACM Transactions on Graphics (TOG)*, 33(4):148, 2014.

[41] Z. Shu, E. Shechtman, D. Samaras, and S. Hadap. Eyeopener: Editing eyes in the wild. *ACM Transactions on Graphics (TOG)*, 36(1):1, 2017.

[42] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras. Neural face editing with intrinsic image disentangling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5541–5550, 2017.

[43] B. M. Smith, L. Zhang, J. Brandt, Z. Lin, and J. Yang. Exemplar-based face parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3484–3491, 2013.

[44] L. Tran, X. Yin, and X. Liu. Disentangled representation learning gan for pose-invariant face recognition. In *CVPR*, volume 3, page 7, 2017.

[45] 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. *arXiv preprint arXiv:1711.11585*, 2017.

[46] H. Yang, D. Huang, Y. Wang, and A. K. Jain. Learning face age progression: A pyramid architecture of gans. *arXiv preprint arXiv:1711.10352*, 2017.

[47] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017.

[48] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, pages 465–476, 2017.