

MaskGAN: Towards Diverse and Interactive Facial Image Manipulation

Cheng-Han Lee¹ Ziwei Liu² Lingyun Wu¹ Ping Luo³

¹SenseTime Research ²The Chinese University of Hong Kong ³The University of Hong Kong

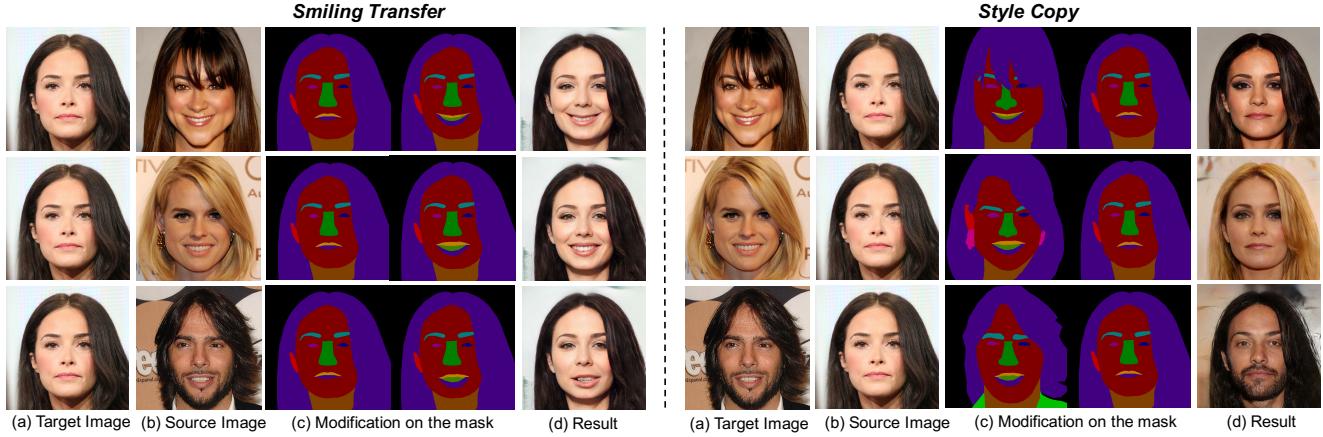


Figure 1: Given a target image (a), users are allowed to modify masks of the target images in (c) according to the source images (b) so that we can obtain manipulation results (d). The left shows illustrative examples from “neutral” to “smiling”, while the right shows style copy such as makeup, hair, expression, skin color, etc.

Abstract

Facial image manipulation has achieved great progresses in recent years. However, previous methods either operate on a predefined set of face attributes or leave users little freedom to interactively manipulate images. To overcome these drawbacks, we propose a novel framework termed MaskGAN, enabling diverse and interactive face manipulation. Our key insight is that semantic masks serve as a suitable intermediate representation for flexible face manipulation with fidelity preservation. MaskGAN has two main components: 1) Dense Mapping Network, and 2) Editing Behavior Simulated Training. Specifically, Dense mapping network learns style mapping between a free-form user modified mask and a target image, enabling diverse generation results. Editing behavior simulated training models the user editing behavior on the source mask, making the overall framework more robust to various manipulated inputs. To facilitate extensive studies, we construct a large-scale high-resolution face dataset with fine-grained mask annotations named CelebAMask-HQ. MaskGAN is comprehensively evaluated on two chal-

lenging tasks: attribute transfer and style copy, demonstrating superior performance over other state-of-the-art methods. The code, models and dataset are available at <https://github.com/switchablenorms/CelebAMask-HQ>.

1. Introduction

Facial image manipulation is an important task in computer vision and computer graphic, enabling lots of applications such as automatic facial expressions and styles (e.g. hair style, skin color) transfer. This task can be roughly categorized into two types: semantic-level manipulation [2, 17, 21, 14, 16] and geometry-level manipulation [30, 29, 31]. However, these methods either operate on a pre-defined set of attributes or leave users little freedom to interactively manipulate the face images.

To overcome the aforementioned drawbacks, we propose a novel framework termed MaskGAN, which aims to enable diverse and interactive face manipulation. Our key insight is that semantic masks serve as a suitable intermediate representation for flexible face manipulation with fidelity preservation. Instead of directly transforming images in the pixel

space, MaskGAN learns the face manipulation process as traversing on the mask manifold [18], thus producing more diverse results with respect to facial components, shapes, and poses. An additional advantage of MaskGAN is that it provides users an intuitive way to specify shape, location, and facial component categories for interactive editing.

MaskGAN has two main components including 1) Dense Mapping Network, and 2) Editing Behavior Simulated Training. The former learns the mapping between the semantic mask and the rendered image, while the latter learns to model the user editing behavior when manipulating masks. Specifically, dense mapping network consists of an image generation backbone and a spatial-aware style encoder. The spatial-aware style encoder takes both target image and its corresponding semantic label mask as inputs; and it produce spatial-aware style features to the image generation backbone. After receiving a source mask with user modification, the image generation backbone leans to synthesize faces according to the spatial-aware style features. In this way, our dense mapping network is capable of learning the fine-grained style mapping between a user modified mask and a target image.

Editing behavior simulated training is a training strategy to model the user editing behavior on the source mask. Its training pipeline comprises an obtained dense mapping network, a pre-trained MaskVAE, and an alpha blender sub-network. Specifically, the MaskVAE with encoder-decoder architecture is responsible for modeling the manifold of geometrical structure priors. The alpha blender sub-network learns to perform alpha blending [22] as image composition, which helps maintaining the manipulation consistency. After training with editing behavior simulation, dense mapping network is more robust to the various changes of the user-input mask during inference.

MaskGAN is comprehensively evaluated on two challenging tasks, including attribute transfer and style copy, showing superior performance compared to other state-of-the-art methods. To facilitate large-scale studies, we construct a large-scale high-resolution face dataset with fine-grained mask labels named CelebAMask-HQ. Specifically, CelebAMask-HQ consists of over 30,000 face images of 512×512 resolution, where each image is annotated with a semantic mask of 19 facial component categories, *e.g.* eye region, nose region, mouth region.

To summarize, our contributions are three-fold: 1) We present MaskGAN for diverse and interactive face manipulation. Within the MaskGAN framework, dense mapping network is further proposed to provide user an interactive way for manipulating face using its semantic label mask. 2) We introduce a novel training strategy termed Editing Behavior Simulated Training, which enhances the robustness of dense mapping network to the shape variations of the user-input mask during inference. 3) We contribute

CelebAMask-HQ, a large-scale high-resolution face dataset with mask annotations. We believe this geometry-oriented dataset would open new research directions for the face editing and manipulation community.

2. Related Work

Image-to-Image Translation. Many recent works have leveraged adversarial training for the image-to-image translation problem [8, 32, 17, 11, 27]. For example, Pix2Pix [8] is the first unified framework based on conditional GAN [20], which used input-output image pairs as training data. In recent years, some studies like [1, 27] start to focus on high-resolution semantic label to image translation. Pix2PixHD [27] extends Pix2Pix [8] to high resolution version by proposing a two stages training framework. These approaches overcame the training instability and optimization issues, when generating high-resolution images. However, they all focused on domain translations and cannot accomplish structure-conditional image manipulation.

Semantic-level Face Manipulation. Deep semantic-level face editing has been surveyed for a few years. Many works including [2, 17, 21, 14, 16] got impressive results. IcGAN [21] introduced an encoder to learn the inverse mappings of conditional GAN. DIAT [16] utilized adversarial loss to transfer attribute and learn to blend predicted face and original face. Fader Network [14] leveraged adversarial training to disentangle attribute related features from the latent space. StarGAN [2] was proposed to perform multi-domain image translation using a single network conditioned on the target domain label. However, these methods cannot generate images by exemplars.

Geometry-level Face Manipulation. Some recent studies [30, 29, 31] start to discuss the possibility of transferring facial at instance level from exemplars. For example, ELEGANT [29] was proposed to exchange attribute between two faces by exchanging the latent codes of two faces. However, ELEGANT [29] cannot transfer the attribute (*e.g.* ‘smiling’) from exemplars accurately.

3. Our Approach

Overall Framework. Our goal is to realize structure conditioned face manipulation. Give an target image $I^t \in \mathbb{R}^{H \times W \times 3}$, a semantic label mask of target image $M^t \in \mathbb{R}^{H \times W \times C}$ and a source semantic label mask $M^{src} \in \mathbb{R}^{H \times W \times C}$ (user modified mask) to MaskGAN. When users manipulating the structure of M^{src} , our model can synthesis a manipulated face $I^{out} \in \mathbb{R}^{H \times W \times 3}$ where C is the category number of semantic label.

Training Pipeline. As shown in Fig. 9, MaskGAN composes of three key elements: Dense Mapping Network (DMN), MaskVAE, and Alpha Blender which are trained by Editing Behavior Simulated Training. Dense Mapping

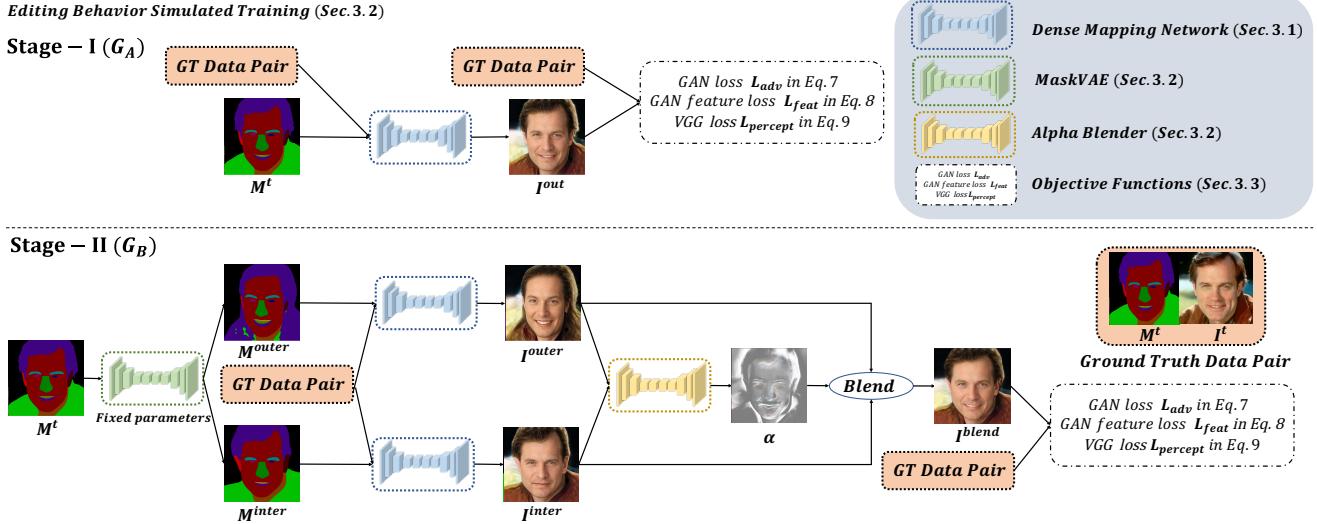


Figure 2: Overall training pipeline. Editing Behavior Simulated Training can be divided into two stage. After loading the pre-trained model of Dense Mapping Network and MaskVAE, we iteratively update these two stages until model converging.

Network (See Sec. 3.1) provides user an interface for manipulating face toward semantic label mask which can learn a style mapping between I^t and M^{src} . MaskVAE is responsible for modeling the manifold of structure priors (See Sec. 3.2). Alpha Blender is responsible for maintaining manipulation consistency (See Sec. 3.2). To make Dense Mapping Network more robust to the changing of the user defined mask M^{src} in inference time, we propose a novel training strategy called Editing Behavior Simulated Training (See Sec. 3.2) which can model the user editing behavior on the M^{src} . This training method need a well trained Dense Mapping Network, a MaskVAE trained until low reconstruction error, and an Alpha Blender trained from scratch. The training pipeline can be divided into two stages. In training stage, we replace M^{src} with M^t as input. In Stage-I, we update DMN with M^t and I^t firstly. In Stage-II, we used MaskVAE to generate two new mask M^{inter} and M^{outer} with small different from M^t and generate two faces I^{inter} and I^{outer} . Then, Alpha Blender blend these two faces to I^{blend} for maintaining manipulation consistency. After Editing Behavior Simulated Training, DMN would be more robust to the change of M^{src} in inference stage. The details of the objective functions are shown in Sec. 3.3.

Inference Pipeline. We only need Dense Mapping Network in testing. In Fig. 10, different from training stage, we simply replace the input of Image Generation Backbone with M^{src} where M^{src} can be defined by the user.

3.1. Dense Mapping Network

Dense Mapping Network adopts the architecture of Pix2PixHD as backbone and we extend it with a external

encoder Enc_{style} which will receive I^t and M^t as inputs. The detailed architecture is shown in Fig. 10.

Spatial-Aware Style Encoder. We propose a Spatial-Aware Style Encoder network Enc_{style} which receives style information I^t and its corresponding spatial information M^t at the same time. To fuse this two domains, we utilize Spatial Feature Transform (SFT) in SFT-GAN [28]. The SFT layer learns a mapping function $\mathcal{M} : \Psi \mapsto (\gamma, \beta)$ where affine transformation parameters (γ, β) is obtained by prior condition Ψ as $(\gamma, \beta) = \mathcal{M}(\Psi)$. After obtaining γ and β , the SFT layer both perform feature-wise and spatial-wise modulation on feature map F as $SFT(F|\gamma, \beta) = \gamma \odot F + \beta$ where the dimension of F is the same as γ and β , and \odot is referred to element-wise product. Here we obtain the prior condition Ψ from the features of M^t and feature map F from I^t . Therefore, we can condition spatial information M^t on style information I^t and generate x_i, y_i as following:

$$x_i, y_i = Enc_{style}(I_i^t, M_i^t), \quad (1)$$

where x_i, y_i are affine parameters which contain spatial-aware style information. To transfer the spatial-aware style information to target mask input, we leverage adaptive instance normalization [6] (AdaIN) on residual blocks z_i in the Dense Mapping Network. The AdaIN operation which is a state-of-the-art method in style transfer is defined as:

$$AdaIN(z_i, x_i, y_i) = x_i \left(\frac{z_i - \mu(z_i)}{\sigma(z_i)} \right) + y_i, \quad (2)$$

which is similar to Instance Normalization [26], but replaces the affine parameters from IN with conditional style information.

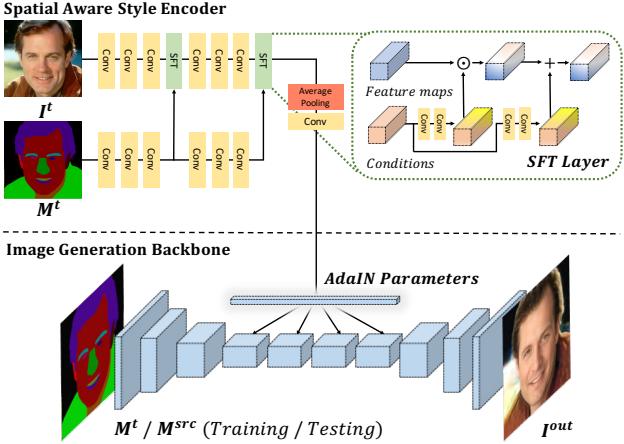


Figure 3: Architecture of Dense Mapping Network which is composed of a **Spatial-Aware Style Encoder** and a **Image Generation Backbone**

Dense Mapping Network is a generator defined as G_A where $I^{out} = G_A(Enc_{style}(I^t, M^t), M^t)$. With the Spatial-Aware Style Encoder, Dense Mapping Network learn the style mapping between I^t and M^{src} according to the spatial information provided by M^t . Therefore, styles (e.g. hair style and skin style) in I^t are transitioned to corresponding position on M^{src} so that DMN can synthesis final manipulated face I^{out} .

3.2. Editing Behavior Simulated Training

Editing Behavior Simulated Training can model the user editing behavior on the M^{src} in training time. This training method need a well trained Dense Mapping Network G_A , a MaskVAE trained until low reconstruction error, and an Alpha Blender trained from scratch. MaskVAE composed of Enc_{VAE} and Dec_{VAE} is responsible for modeling the manifold of structure priors. Alpha Blender B is responsible for maintaining manipulation consistency. We define G_B as another generator which utilize MaskVAE, Dense Mapping Network, and Alpha Blender as G_B where $G_B \equiv B(G_A(I^t, M^t, M^{inter}), G_A(I^t, M^t, M^{outer}))$. The overall training pipeline is shown in Fig. 9 and the detailed algorithm is shown in Algo. 1. Our training pipeline can be divided into two stages. Firstly, we need to load pretrained model of G_A , Enc_{VAE} and Dec_{VAE} . In stage-I, we update G_A once. In stage-II, given M^t , we obtain two new masks M^{inter} and M^{outer} with small structure interpolation and extrapolation from original one by adding two parallel vectors with reverse direction on the latent space of the mask. These vectors are obtained by $\pm \frac{z^{ref} - z^t}{\lambda_{inter}}$ where z^{ref} is latent representation of a random selected mask M^{ref} and λ_{inter} is set to 2.5 for appropriate blending. After generating two

Algorithm 1 Editing Behavior Simulated Training

Initialization: Pre-trained G_A , Enc_{VAE} , Dec_{VAE} models
Input: I^t, M^t, M^{ref}
Output: I^{out}, I^{blend}

- 1: **while** iteration not converge **do**
- 2: Choose one minibatch of N mask and image pairs $\{M_i^t, M_i^{ref}, I_i^t\}$, $i = 1, \dots, N$.
- 3: $z^t = Enc_{VAE}(M^t)$
- 4: $z^{ref} = Enc_{VAE}(M^{ref})$
- 5: $z^{inter}, z^{outer} = z^t \pm \frac{z^{ref} - z^t}{\lambda_{inter}}$
- 6: $M^{inter} = Dec_{VAE}(z^{inter})$
- 7: $M^{outer} = Dec_{VAE}(z^{outer})$
- 8: Update $G_A(I^t, M^t)$ with Eq. 6
- 9: Update $G_B(I^t, M^t, M^{inter}, M^{outer})$ with Eq. 6
- 10: **end while**

faces by Dense Mapping Network, Alpha Blender learns to blend two images toward target image where keeping the consistency with the original one. Then, we iteratively update the G_A and G_B (*Stage – I* and *Stage – II* in Fig. 9) until model converging. After Editing Behavior Simulated Training, Dense Mapping Network would be more robust to the change of the user modified mask in inference time.

Structural Priors by MaskVAE. Similar to Variational Autoencoder [13], the objective function for learning a MaskVAE consists of two parts: (i) $L_{reconstruct}$, which controls the pixel-wise semantic label difference, (ii) L_{KL} , which controls the smoothness in the latent space. The overall objective is to minimize the following loss function:

$$\mathcal{L}_{MaskVAE} = \mathcal{L}_{reconstruct} + \lambda_{KL} \mathcal{L}_{KL}, \quad (3)$$

where λ_{KL} is set to $1e^{-5}$ which is obtained through cross validation. The encoder network $Enc_{VAE}(M^t)$ output the mean μ and covariance σ of the latent vector. We use KL divergence loss to minimize the gap between the prior $P(z)$ and the learned distribution, i.e.

$$\mathcal{L}_{KL} = \frac{1}{2}(\mu\mu^T + \sum_{j=1}^J(exp(\sigma) - \sigma - 1)), \quad (4)$$

where denotes the j -th element of vector σ . Then, we can sample latent vector by $z = \mu + r \odot exp(\sigma)$ in the training phase, where $r \sim N(0, I)$ is a random vector and \odot denotes element-wise multiplication.

The decoder network $Dec_{VAE}(z)$ output the reconstruct semantic label and calculate pixel-wise cross-entropy loss as following:

$$\mathcal{L}_{reconstruct} = -\mathbb{E}_{z \sim P(z)}[\log(P(M^t|z))]. \quad (5)$$

Fig. 11 shows some samples of linear interpolation between two mask. MaskVAE can perform smooth transition on masks.

Manipulation Consistency by Alpha Blender. To maintain the consistency of manipulation between I^{blend} and I^t ,

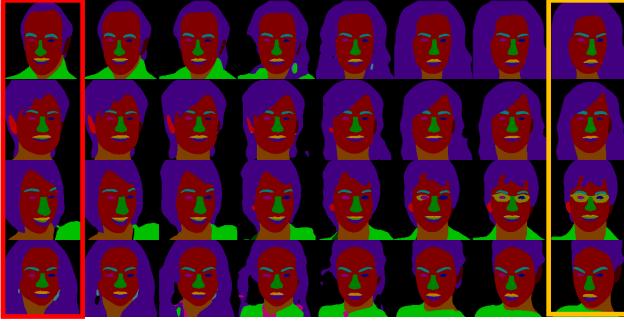


Figure 4: Samples of linear interpolation between two masks (between the red block and the orange block). MaskVAE can perform smooth transition on masks.

we realize alpha blending [22] used in image composition by a deep neural network based Alpha Blender B which learn the alpha blending weight α with two input images : I^{inter} and I^{outer} as $\alpha = B(I^{inter}, I^{outer})$. After learning appropriated α , Alpha Blender blend I^{inter} and I^{outer} according $I^{blend} = \alpha \times I^{inter} + (1 - \alpha) \times I^{outer}$. As shown in the *Stage - II* of Fig. 9, Alpha Blender is jointly optimized with two share weighted Dense Mapping Networks. The group of models are defined as G_B .

3.3. Multi-Objective Leaning

The objective function for learning both G_A and G_B consists of three parts: (i) L_{adv} , which is the conditional adversarial loss that make generated images more realistic and correct the generation structure according to the conditional mask M^t , (ii) L_{feat} , which encourages generator to produce natural statistic at multiple scales, (iii) $L_{percept}$, which improves content generation from low frequency to high frequency details in perceptually toward deep features in VGG-19 [25] trained by ImageNet [3]. To improve synthesis quality of high-resolution image, we leverage multi-scale discriminator [27] to increase the receptive field and decrease repeated patterns appearing in the generated image. We used two discriminators which refer to $D_{1,2}$ with identical network structure to operate at two different scales. The overall objective is to minimize the following loss function.

$$\begin{aligned} \mathcal{L}_{G_A, G_B} &= \mathcal{L}_{adv}(G, D_{1,2}) \\ &+ \lambda_{feat} \mathcal{L}_{feat}(G, D_{1,2}) \\ &+ \lambda_{percept} \mathcal{L}_{percept}(G), \end{aligned} \quad (6)$$

where λ_{feat} and $\lambda_{percept}$ are set to 10 which are obtained through cross validation.

\mathcal{L}_{adv} is the conditional adversarial loss defined by

$$\mathcal{L}_{adv} = \mathbb{E}[\log(D_{1,2}(I^t, M^t))] + \mathbb{E}[1 - \log(D_{1,2}(I^{out}, M^t))]. \quad (7)$$

\mathcal{L}_{feat} is the feature matching loss [27] which computes the $L1$ distance between the real and generated image using

	Helen [15]	CelebAMask-HQ
# of Images	2.33K	30K
Mask size	400×600	512×512
# of Categories	11	19

Table 1: Dataset statistics comparisons with existing dataset. CelebAMask-HQ has superior scales on the number of images and also category annotations.

the intermediate features from discriminator by

$$\mathcal{L}_{feat} = \mathbb{E} \sum_{i=1} \|D_{1,2}^{(i)}(I^t, M^t) - D_{1,2}^{(i)}(I^{out}, M^t)\|_1. \quad (8)$$

$\mathcal{L}_{percept}$ is the perceptual loss [9] which computes the $L1$ distance between the real and generated image using the intermediate features from a fixed VGG-19 [25] model by

$$\mathcal{L}_{percept} = \sum_{i=1} \frac{1}{M_i} [\|\phi^{(i)}(I^t) - \phi^{(i)}(I^{out})\|_1]. \quad (9)$$

4. CelebAMask-HQ Dataset

We build a large-scale face semantic label dataset named CelebAMask-HQ, which is labeled according to CelebA-HQ [10] that contains 30,000 high-resolution face images from CelebA [19]. It has several appealing properties:

- **Comprehensive Annotations.** CelebAMask-HQ was precisely hand-annotated with the size of 512×512 and 19 classes including all facial components and accessories such as ‘skin’, ‘nose’, ‘eyes’, ‘eyebrows’, ‘ears’, ‘mouth’, ‘lip’, ‘hair’, ‘hat’, ‘eyeglass’, ‘ear-ring’, ‘necklace’, ‘neck’, and ‘cloth’.
- **Amodal Handling.** For occlusion handling, if the facial component was partly occluded, we ask annotators to label the occluded parts of the components by human inferring. On the other hand, we skip the annotations for those components that are totally occluded.

Table 1 compares the dataset statistics of CelebAMask-HQ with Helen dataset [15].

5. Experiments

We comprehensively evaluate our approach by showing quantitative and visual quality on different benchmarks.

5.1. Datasets

CelebA-HQ. [10] is a high quality facial image dataset that consist of 30000 images picked from CelebA dataset [19]. These images are processed with quality improvement to the size of 1024×1024 . We resize all images to the size of 512×512 for our experiments.

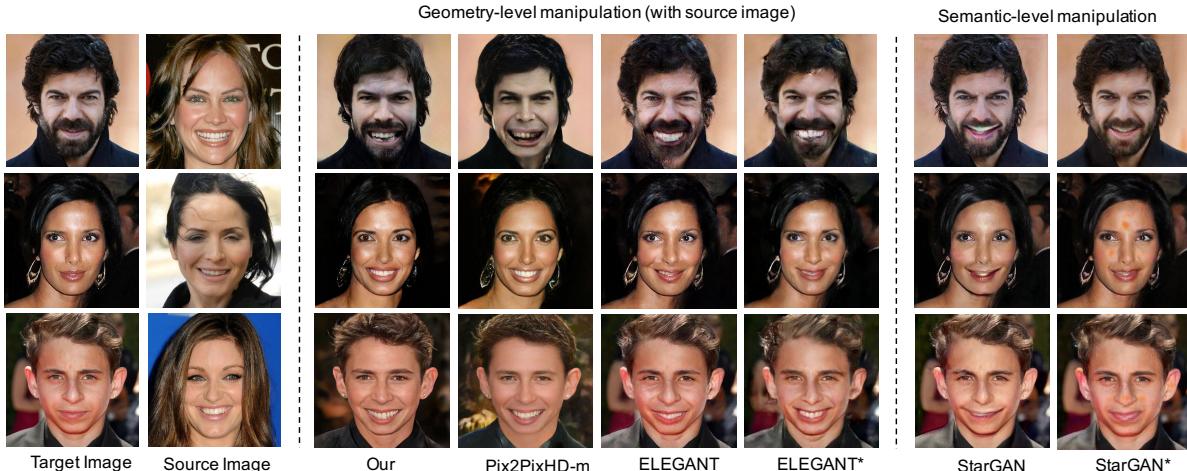


Figure 5: Visual comparison with other methods for specific attribute: **Smiling** on facial attribute transfer. * means the model is trained by images with the size of 256×256 . The first two columns are target and source pairs. The middle four columns show the results of geometry-level manipulation (our MaskGAN, Pix2PixHD-m [27], and ELEGANT [29]) which utilize source images as exemplars. The last two columns show the results based on semantic-level manipulation (e.g. StarGAN [2]). StarGAN failed on the region of smiling. ELEGANT has plausible results, but sometimes cannot transfer smiling from the source image accurately. Pix2PixHD-m has lower perceptual quality than others. Our MaskGAN has plausible visual quality and relatively better geometry-level smiling transfer ability.

Metric	Attribute cls. accuracy(%)	Segmentation(%)	FID score	Human eval.(%)
StarGAN* [2]	62.8	-	40.57	-
StarGAN [2]	57.0	-	31.91	5
ELEGANT* [29]	67.3	-	50.54	-
ELEGANT [29]	64.0	-	37.20	29
Pix2PixHD-m [27]	80.5	92.9	54.41	30
MaskGAN	73.3	92.5	48.24	-
MaskGAN [†]	78.0	93.0	48.76	36
GT	89.0	92.1	-	-

Table 2: Evaluation on geometry-level facial attribute transfer. Quantitative comparison with other methods for specific attribute - **Smiling**. * means the model is trained by images with the size of 256×256 . † means the model is trained by **Editing Behavior Simulated Training**. StarGAN and ELEGANT have better FID scores, but lower attribute classification accuracy. Pix2PixHD-m get the best classification accuracy, but has inferior FID scores than others. Although MaskGAN cannot achieve the best FID score, it has relatively higher classification accuracy and segmentation accuracy.

CelebAMask-HQ. Based on CelebA-HQ, we propose a new dataset named CelebAMask-HQ which has 30000 semantic segmentation labels with the size of 512×512 . Each label in the dataset has 19 classes.

5.2. Implementation Details

Training Details. Our pre-trained Dense Mapping Network and MaskVAE are both updated with the Adam optimizer [12] ($\beta_1 = 0.5$, $\beta_2 = 0.999$, learning rate of $2e^{-4}$) For Editing Behavior Simulated Training, we reduce the learning rate to $5e^{-5}$. MaskVAE is trained with batch size of 16 and MaskGAN is trained with the batch size of 8.

Network Architectures. Image Generation Backbone in Dense Mapping Network follows the design of Pix2PixHD [27] with 4 residual blocks. Alpha Blender also follows the design of Pix2PixHD but only downsampling 3 times

and using 3 residual blocks. The architecture of MaskVAE is similar to UNet [23] without skip-connection. Spatial-Aware Style Encoder in Dense Mapping Network dose not use any Instance Normalization [26] layers which will remove style information. All the other convolutional layers in Dense Mapping Network, Alpha Blender, and Discriminator are followed by IN layers. MaskVAE utilizes Batch Normalization [7] in all layers.

Competing Methods. We choose state-of-the-art StarGAN [2], ELEGANT [29] and Pix2PixHD [27] as our baselines. StarGAN performs semantic-level facial attribute manipulation. ELEGANT performs geometry-level facial attribute manipulation. Pix2PixHD performs photo-realistic image synthesis from semantic mask. We simply remove the branch for receiving M^t in Spatial-Aware Style Encoder of Dense Mapping Network as baseline called Pix2PixHD-m.



Figure 6: Visual comparison with other methods on style copy. All the columns show the results of the proposed method and Pix2PixHD-m [27] for four different target images. MaskGAN shows better ability to transfer style like makeup and gender than Pix2PixHD-m.

Metric	Attribute cls. accuracy(%)			Segmentation(%)	FID score	Human eval.(%)
Pix2PixHD-m [27]	57.9	54.8	78.4	91.1	40.63	24
MaskGAN	72.0	71.4	88.2	91.6	38.15	25
MaskGAN [†]	75.0	73.5	89.1	91.5	38.01	51
GT	96.4	89.1	95.4	92.9	-	-

Table 3: Evaluation on geometry-level style copy. Quantitative comparison with other methods. [†] means the model is trained by **Editing Behavior Simulated Training**. Attribute types in attribute classification accuracy from left to right are **Male**, **Heavy Makeup**, and **No Beard**. MaskGAN has relatively high attribute classification accuracy than Pix2PixHD-m. **Editing Behavior Simulated Training** further improves the robustness of attribute keeping ability so that MaskGAN[†] has higher attribute classification accuracy and human evaluation score than MaskGAN.

5.3. Evaluation Metrics

Semantic-level Evaluation. To evaluate a method of manipulating a target attribute, we examined the classification accuracy of synthesized images. We trained binary facial attribute classifiers for specific attributes on the CelebA dataset by using ResNet-18 [4] architecture.

Geometry-level Evaluation. To measure the quality of mask-conditional image generation, we apply a pre-trained a face parsing model with U-Net [23] architecture to the generated images, and measure the consistency between the input layout and the predicted parsing results in terms of pixel-wise accuracy.

Distribution-level Evaluation. To measure the quality of generated images from different models. We use the Frchet Inception Distance [5] (FID) to measure the quality and diversity of generated images.

Human Perception Evaluation. We perform a user survey to evaluate perceptual generation quality. Given a target image (and a source image in the experiment of style copy), the

user was required to choose the best generated image based on two criteria: 1) quality of transfer in attributes and style 2) perceptual realism. The options are randomly shuffled images generated from different methods.

5.4. Comparisons with Prior Works

The comparison is performed w.r.t. three aspects, including semantic-level evaluation, geometry-level evaluation, and distributed-level evaluation. We denote our approach as MaskGAN and MaskGAN[†] for reference, where [†] means the model uses Editing Behavior Simulated Training. For Pix2PixHD [27] with modification, we name it as Pix2PixHD-m for reference.

Evaluation on Attribute Transfer. We choose **Smiling** to compare which is the most challenging attribute type to transfer in previous works. To generate the user modified mask as input, we conduct head pose estimation on the testing set by using the HopeNet [24]. With the angle information of roll, pitch, and yaw, we selected 400 source and

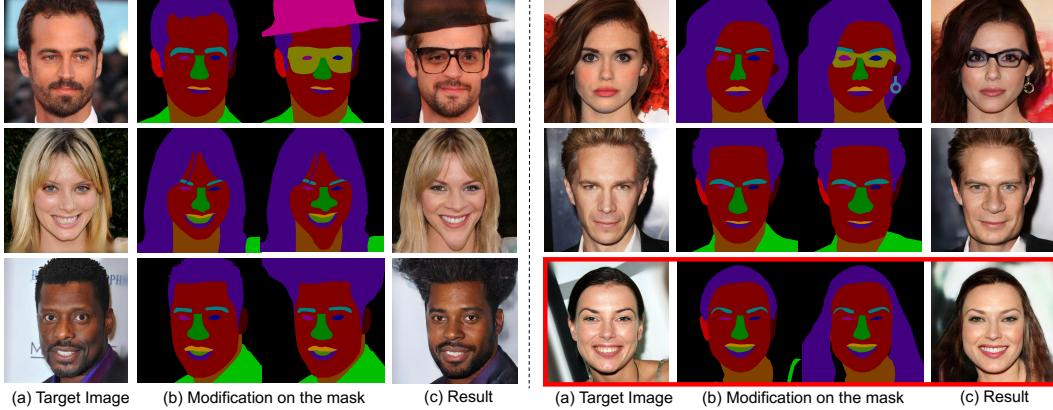


Figure 7: Visual results of interactive face editing. The first row shows examples of adding accessories like eyeglass, earring, and hat. The second row shows examples of editing face shape and nose shape. The third row show examples of adding hair. The red block shows an fail case that the strength of the hair color may decrease when adding hair to a short hair woman.

target pairs with similar pose from the testing set. Then, we directly replace the mask of mouth, upper lip and lower lip from target mask to source mask. Fig. 12 and Table 2 show the visual results and quantitative results on MaskGAN and state-of the art. For fair comparison, StarGAN* and ELEGANT* mean model trained by images with the size of 256×256 . StarGAN has the best FID scores, but failed on the region of smiling for the reason that the performance of StarGAN may be influenced by the size of the training data and network design. ELEGANT has plausible results, but sometimes cannot transfer smiling from the source image accurately because it exchange attribute from source image in latent space. Pix2PixHD-m gets the best classification accuracy, but has inferior FID scores than others. As long as the target image does not have spatial information to learn a better mapping with the user defined mask. Although MaskGAN cannot achieve the best FID score, it has plausible visual quality and relative high classification accuracy and segmentation accuracy.

Evaluation on Style Copy. To illustrate the robustness of our model, we test MaskGAN on a more difficult task: geometry-level style copy. Style copy can also be seen as manipulating a face structure to another face. We select 1000 target images from testing set and the source images are selected from the target images with totally different order. For this setting, about half of pairs are different gender. Fig. 6 and Table 3 show the visual results and quantitative results on MaskGAN and state-of the art. From the visual results and attribute classification accuracy (from left to right : **Male, Heavy Makeup, and No Beard**), MaskGAN shows relatively better ability to transfer style like makeup and gender than Pix2PixHD-m, because MaskGAN introduces spatial information to the style features and simulate the user editing behavior in training.

5.5. Ablation Study

In the ablation study, we consider two variants of our model: (i) MaskGAN and (ii) MaskGAN[†].

Dense Mapping Network. In Fig. 6, we observe that Pix2PixHD-m is influenced by the prior information contained in the user modified mask. For example, if the user modified the mask to be a female while the target image looks like a male, the predicted image tends to a female with makeup and no beard. Besides, Pix2PixHD-m cannot transition the style from the target image to the user modified mask accurately. With Spatial-Aware Style Encoder, MaskGAN not only prevents generated results influenced by prior knowledge in user modified mask, but also accurately transfers the style of the target image.

Editing Behavior Simulated Training. Table 2 and Table 3 show that simulating editing behavior in training can prevent content generation in inference stage from being influenced by structure changing on the user modified mask. It improves the robustness of attribute keeping ability so that MaskGAN has better human evaluation score.

5.6. Real-life Applications

Interactive Face Editing. Our MaskGAN allows users to interactively edit the shape, location, and category of facial components at geometry-level through a semantic mask interface. The interactive face editing results are illustrated in Fig. 7. The first row shows examples of adding accessories like eyeglass, earring, and hat. The second row shows examples of editing face shape and nose shape. The third row show examples of adding hair.

Failure Cases. The red block shows a typical failure case, where the strength of the hair color may decrease when adding hair to a short hair woman. It is due to the large discrepancy between style transition here.

6. Conclusions

In this work, we have proposed a novel geometry-oriented face manipulation framework, MaskGAN, with two carefully designed components: 1) Dense Mapping Network, and 2) Editing Behavior Simulated Training. Our key insight is that semantic masks serve as a suitable intermediate representation for flexible face manipulation with fidelity preservation. MaskGAN is comprehensively evaluated on two challenging tasks: attribute transfer and style copy, showing superior performance over other state-of-the-art methods. We further contribute a large-scale high-resolution face dataset with fine-grained mask annotations, named CelebAMask-HQ. Future work includes combining MaskGAN with image completion techniques to further preserve details on the regions without editing.

References

- [1] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *ICCV*, 2017. [2](#)
- [2] 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. In *CVPR*, 2018. [1](#), [2](#), [6](#), [10](#)
- [3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. [5](#)
- [4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. [7](#)
- [5] 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 *NIPS*, 2017. [7](#)
- [6] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. [3](#), [10](#)
- [7] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. [6](#)
- [8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017. [2](#), [10](#)
- [9] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. [5](#)
- [10] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. [5](#)
- [11] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017. [2](#)
- [12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. [6](#)
- [13] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. [4](#)
- [14] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. De-noyer, et al. Fader networks: Manipulating images by sliding attributes. In *NIPS*, 2017. [1](#), [2](#)
- [15] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *ECCV*, 2012. [5](#)
- [16] M. Li, W. Zuo, and D. Zhang. Deep identity-aware transfer of facial attributes. *arXiv preprint arXiv:1610.05586*, 2016. [1](#), [2](#)
- [17] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *NIPS*, 2017. [1](#), [2](#)
- [18] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *ICCV*, 2015. [2](#)
- [19] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. [5](#)
- [20] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. [2](#), [10](#)
- [21] G. Perarnau, J. Van De Weijer, B. Raducanu, and J. M. Alvarez. Invertible conditional gans for image editing. *arXiv preprint arXiv:1611.06355*, 2016. [1](#), [2](#)
- [22] T. Porter and T. Duff. Compositing digital images. In *ACM Siggraph Computer Graphics*, volume 18, pages 253–259. ACM, 1984. [2](#), [5](#)
- [23] 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*. Springer, 2015. [6](#), [7](#), [10](#)
- [24] N. Ruiz, E. Chong, and J. M. Rehg. Fine-grained head pose estimation without keypoints. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018. [7](#)
- [25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [5](#)
- [26] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. [3](#), [6](#)
- [27] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. [2](#), [5](#), [6](#), [7](#), [10](#)
- [28] X. Wang, K. Yu, C. Dong, and C. C. Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *CVPR*, 2018. [3](#), [10](#)
- [29] T. Xiao, J. Hong, and J. Ma. Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. *arXiv preprint arXiv:1803.10562*, 2018. [1](#), [2](#), [6](#), [10](#)
- [30] R. Yeh, Z. Liu, D. B. Goldman, and A. Agarwala. Semantic facial expression editing using autoencoded flow. *arXiv preprint arXiv:1611.09961*, 2016. [1](#), [2](#)
- [31] W. Yin, Z. Liu, and C. C. Loy. Instance-level facial attributes transfer with geometry-aware flow. *arXiv preprint arXiv:1811.12670*, 2018. [1](#), [2](#)
- [32] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017. [2](#)

A. Additional Implementation Details

Our MaskGAN is composed of four key components: MaskVAE, Dense Mapping Network, Alpha Blender, and Discriminator. Specifically, Dense Mapping Network contains two elements: Image Generation Backbone, Spatial-Aware Style Encoder. More details about architecture design of these components are shown as below.

MaskVAE. The architecture of MaskVAE is similar to UNet [23] without skip-connection. Detailed architectures of *Enc_{VAE}* and *Dec_{VAE}* are shown in Fig. 8 which uses BN for all layers.

Image Generation Backbone. We choose the architecture of Pix2PixHD [27] as Image Generation Backbone. The detailed architecture is as following:

$c7s1 - 64, d128, d256, d512, d1024, R1024, R1024, R1024, u512, u256, u128, u64 - c7s1$.

We utilize AdaIN [6] for all residual blocks, other layers use IN. We do not further utilize local enhancer because we conduct all experiments with image size of 512×512 .

Spatial-Aware Style Encoder. As shown in Fig. 9, Spatial-Aware Style Encoder consists of two branches for receiving both style and spatial information. Two fuse two different domains, we leverage SFT Layers in SFT-GAN [28]. The detailed architecture of SFT Layer is shown in Fig. 10 which does not use any normalization for all layers.

Alpha Blender. Alpha Blender also follows the design of Pix2PixHD but only downsampling three times and using three residual blocks. The detailed architecture is as following:

$c7s1 - 32, d64, d128, d256, R256, R256, R256, u128, u64, u32 - c7s1$ which uses IN for all layers.

Discriminator. Our design of discriminator also follows Pix2PixHD [27] which utilize PatchGAN [8]. We concatenate the masks and images as inputs to realize conditional GAN [20]. The detailed architecture is as following: $c64, c128, c256, c512$ which uses IN for all layers.

B. Additional Visual Results

In Fig. 11, we show additional visual results of attribute transfer for specific attribute: **Smiling**. We compare our MaskGAN with state-of-the-art methods including Pix2PixHD [27] with modification, ELEGANT [29], and StarGAN [2].

In Fig. 12, we show additional visual results of style. We compare our MaskGAN with state-of-the-art methods including Pix2PixHD [27] with modification.

In the accompanying [video](#), we demonstrate our interactive facial image manipulation interface. Users can edit shape of facial components or add some accessories toward manipulating the semantic segmentation mask.

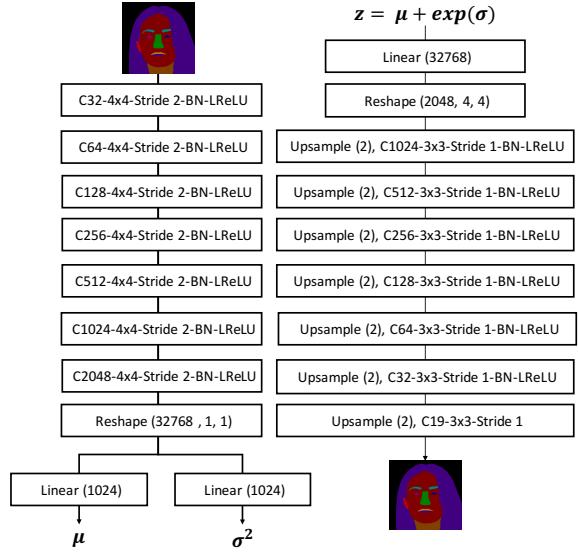


Figure 8: Architecture of MaskVAE.

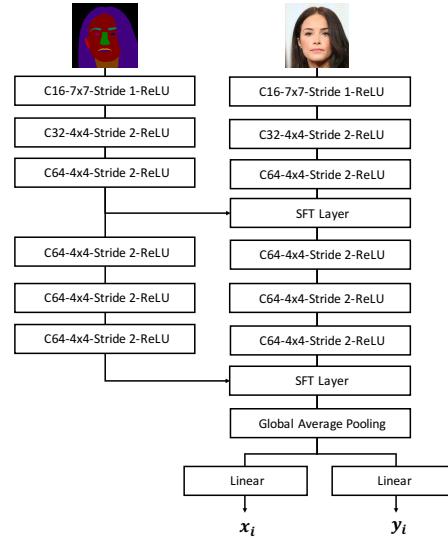


Figure 9: Architecture of Spatial-Aware Style Encoder.

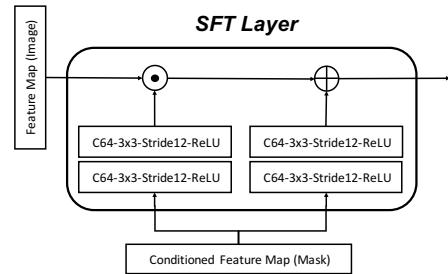


Figure 10: Architecture of Spatial Feature Transform Layer.

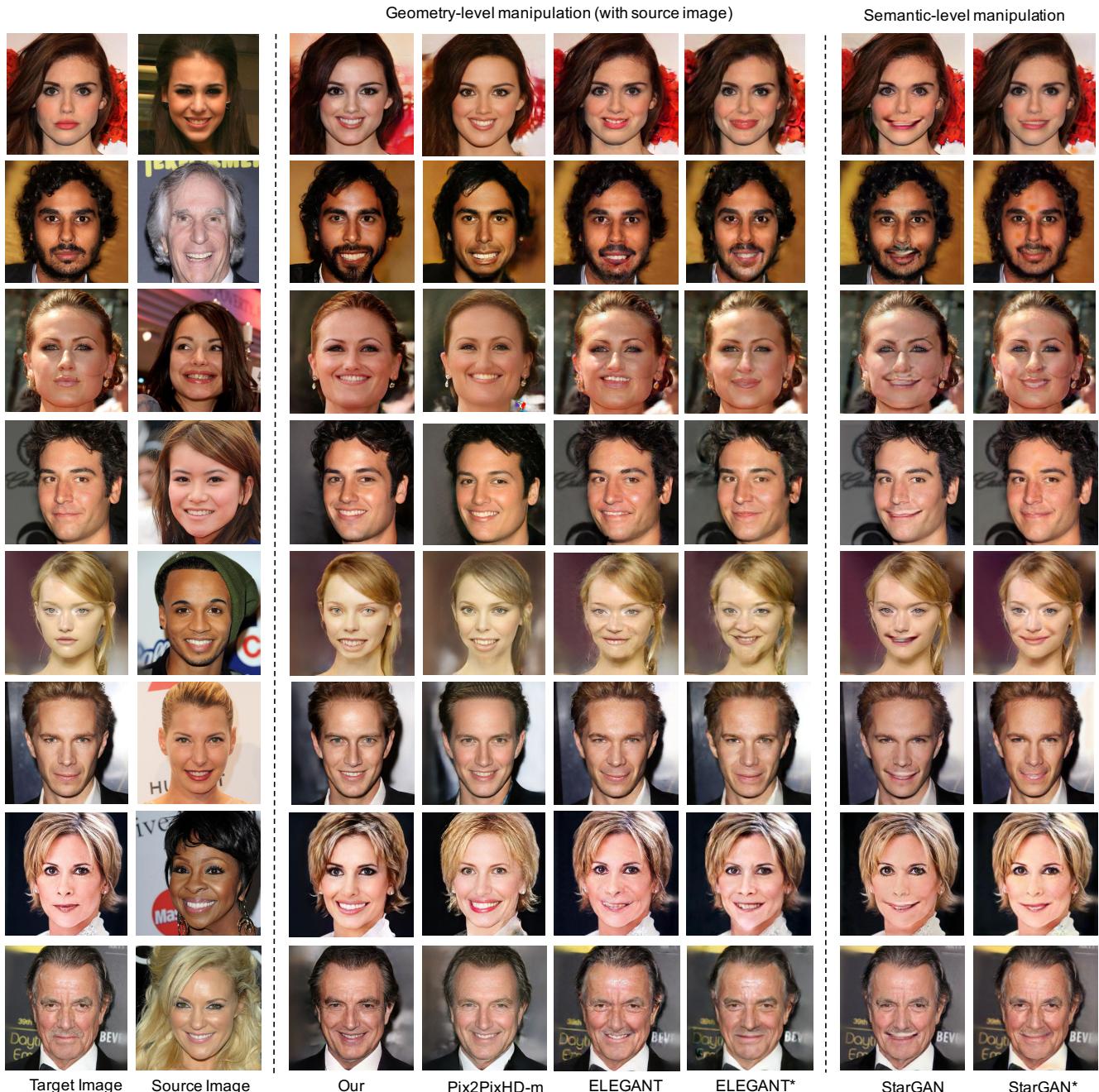


Figure 11: Visual results of attribute transfer for specific attribute: **Smiling**. * means the model is trained with the size of 256×256 .

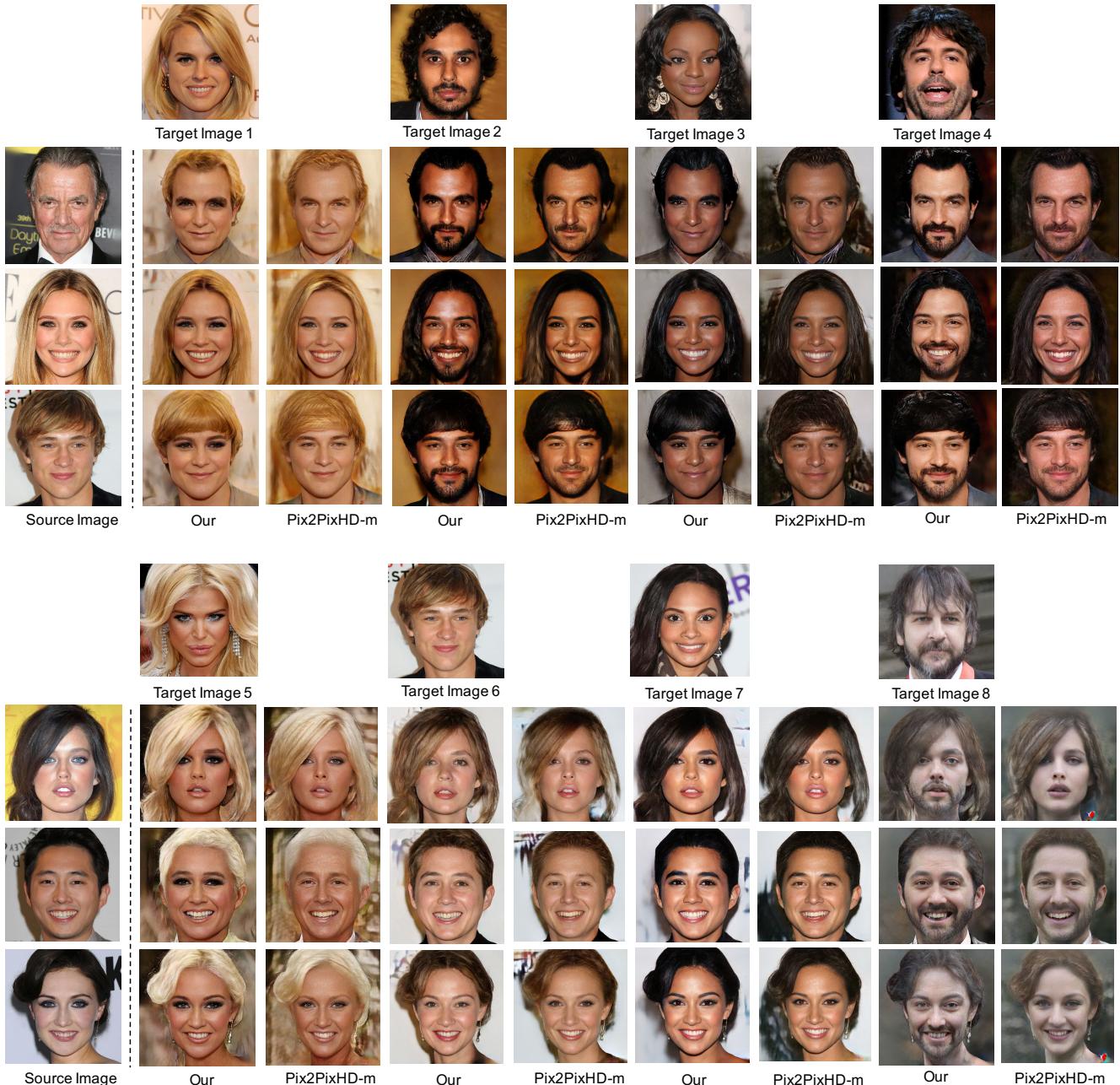


Figure 12: Visual results of style copy.