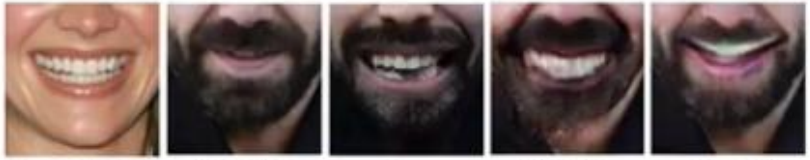
MaskGAN:

Towards Diverse and Interactive Facial Image Manipulation

SenseTime Research, Hong Kong Univ, CVPR 2020
Cheng-Han Lee et al. / Presenter : Taeu

Previous problem

Smiling transfer



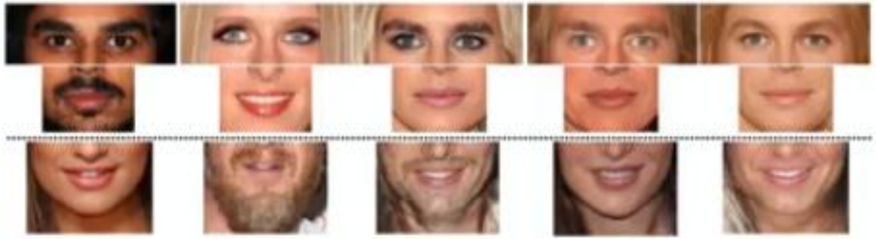
Source Image Target Image Our ELEGANT StarGAN

ELEGANT (Xiao et. al, ECCV 2018)

StarGAN (Chou et. al, CVPR 2018)

- ELEGANT cannot synthesis well when the mouth opens largely.
- StarGAN fails on high-resolution (e.g. 512 x 512) images.

Style Copy



Source Image Target Image Our SPADE Pix2PixHD-m

SPADE (Park et. al, CVPR 2019)

Pix2PixHD (Wang et. al, CVPR 2018)

- SPADE and Pix2PixHD-m cannot preserve the attributes well in inference time.

Goal

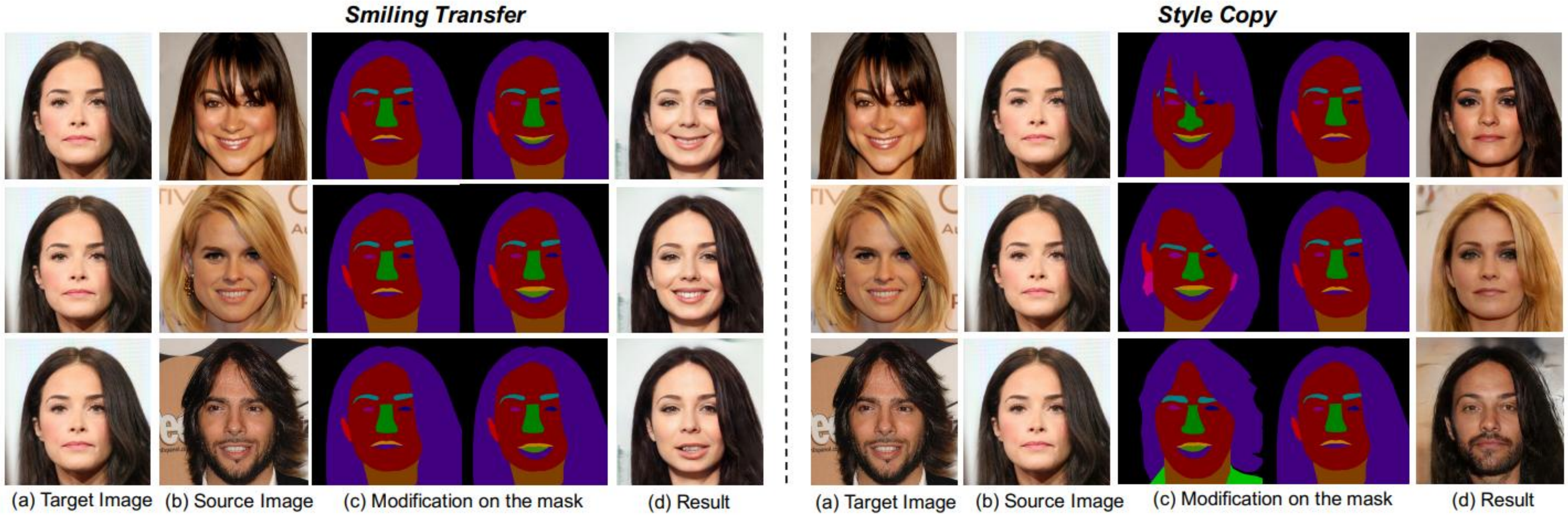
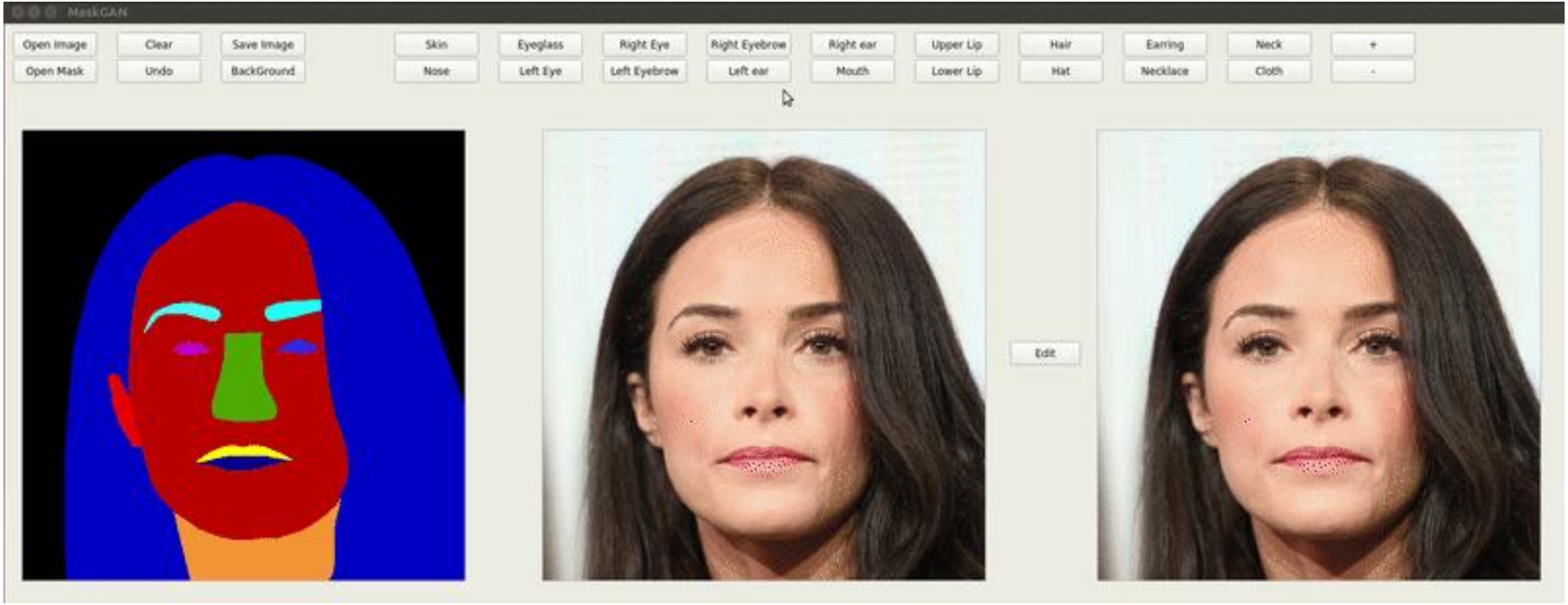


Figure 1: Given a target image (a), users are allowed to modify masks of the target images in (c) according to the source images (b) so that we can obtain manipulation results (d). The left shows illustrative examples from “neutral” to “smiling”, while the right shows style copy such as makeup, hair,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

Demo


MaskGAN
Open Image
Clear
Save Image
Skin
Eyeglass
Right Eye
Right Eyebrow
Right ear
Upper Lip
Hair
Earring
Neck
+
Open Mask
Undo
BackGround
Nose
Left Eye
Left Eyebrow
Left ear
Mouth
Lower Lip
Hat
Necklace
Cloth
-
Edit

Framework

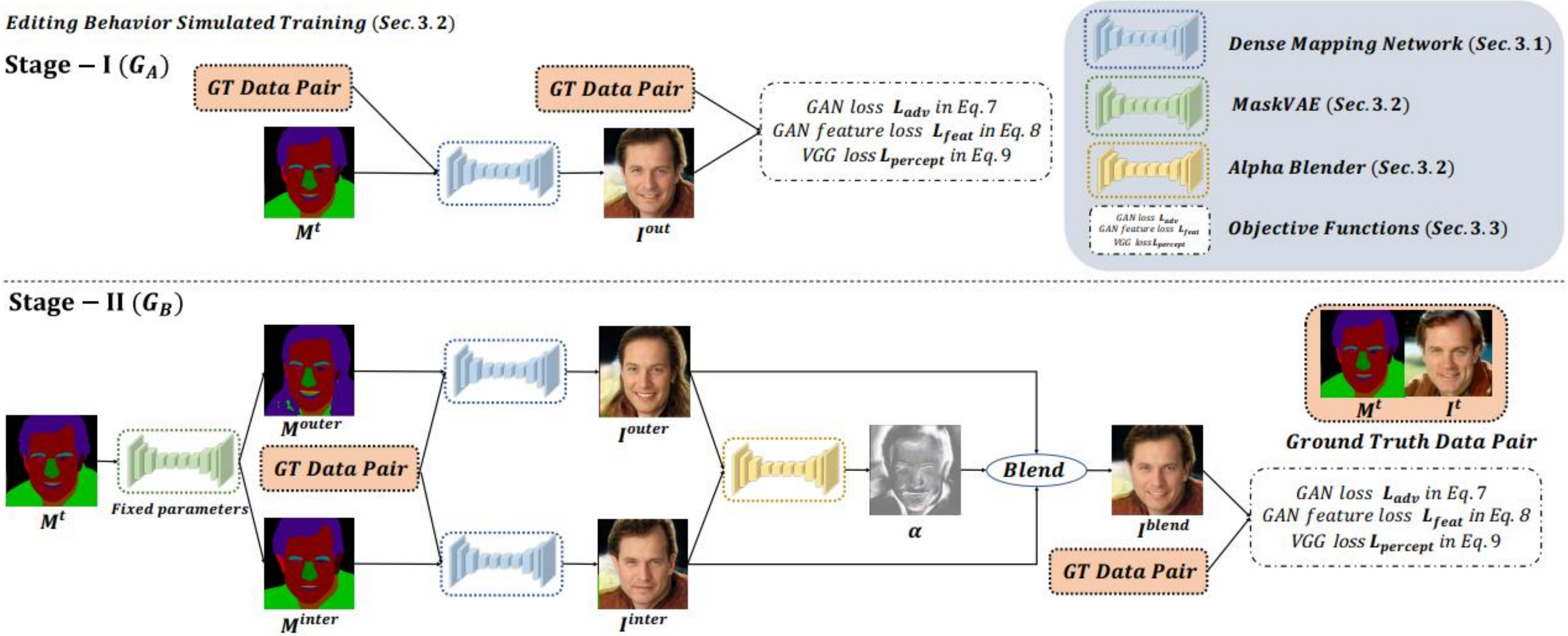

Editing Behavior Simulated Training (Sec. 3. 2)
Stage – I (G_A)
GT Data Pair
GT Data Pair
M^t
I^{out}
GAN loss L_{adv} in Eq. 7
GAN feature loss L_{feat} in Eq. 8
VGG loss $L_{percept}$ in Eq. 9
Dense Mapping Network (Sec. 3. 1)
MaskVAE (Sec. 3. 2)
Alpha Blender (Sec. 3. 2)
GAN loss L_{adv}
GAN feature loss L_{feat}
VGG loss $L_{percept}$
Objective Functions (Sec. 3. 3)
Stage – II (G_B)
M^t
Fixed parameters
M^{outer}
GT Data Pair
M^{inter}
I^{outer}
I^{inter}
α
Blend
I^{blend}
GT Data Pair
M^t
I^t
Ground Truth Data Pair
GAN loss L_{adv} in Eq. 7
GAN feature loss L_{feat} in Eq. 8
VGG loss $L_{percept}$ in Eq. 9

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.

Dense Mapping Network with AdaIN

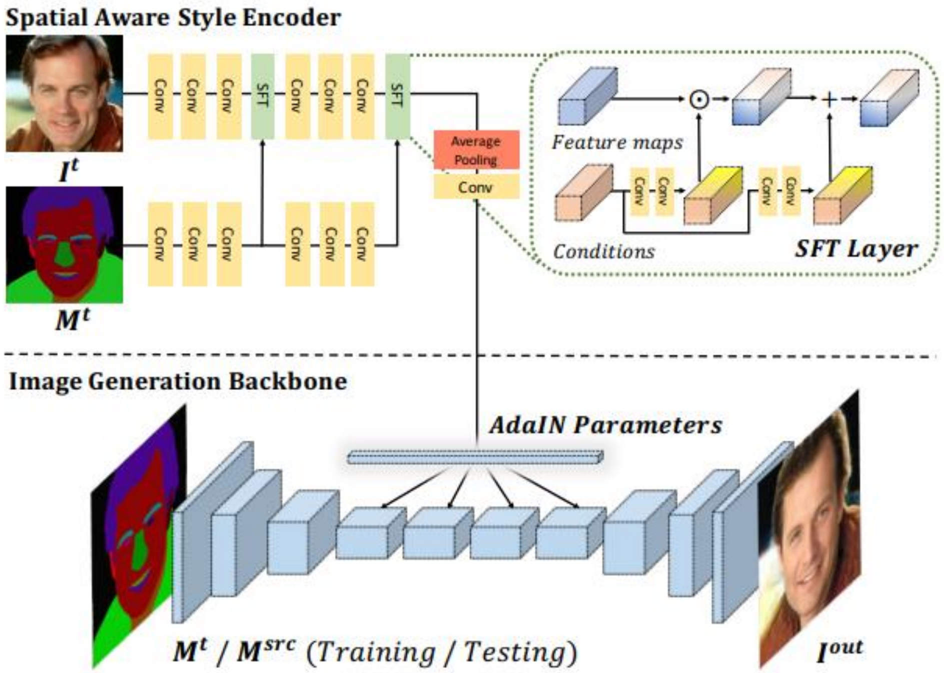


Figure 3: Architecture of Dense Mapping Network which is composed of a **Spatial-Aware Style Encoder** and a **Image Generation Backbone**.

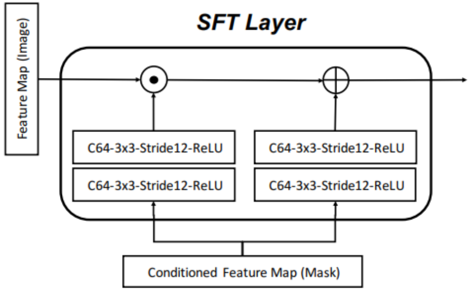


Figure 12: Architecture of Spatial Feature Transform Layer.

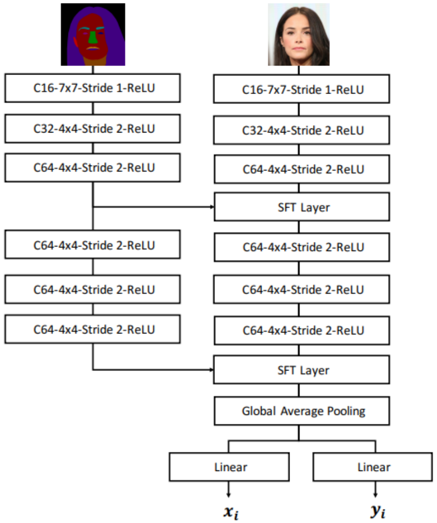


Figure 11: Architecture of Spatial-Aware Style Encoder.

$$x_i, y_i = Enc_{style}(I_i^t, M_i^t), \quad AdaIN(z_i, x_i, y_i) = x_i\left(\frac{z_i - \mu(z_i)}{\sigma(z_i)}\right) + y_i,$$

Framework - Again

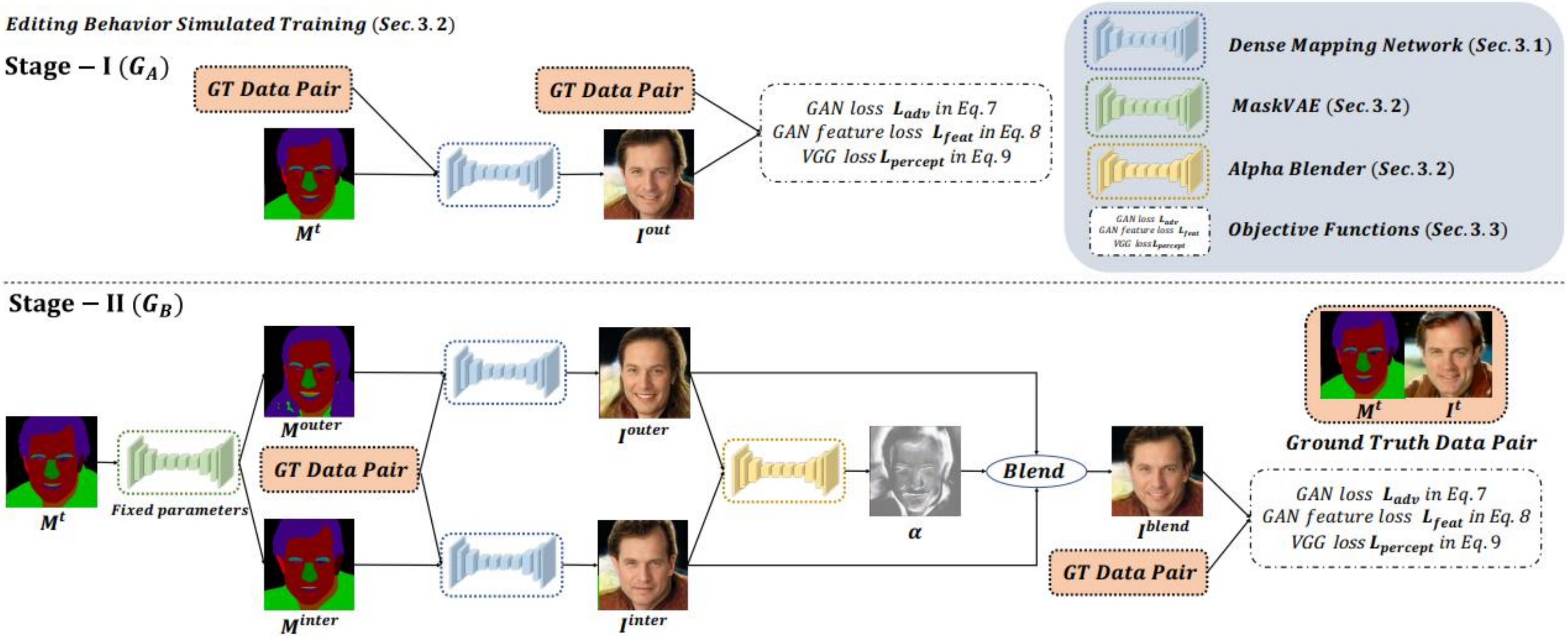

Editing Behavior Simulated Training (Sec. 3. 2)
Stage – I (G_A)
GT Data Pair
GT Data Pair
M^t
I^{out}
GAN loss L_{adv} in Eq. 7
GAN feature loss L_{feat} in Eq. 8
VGG loss $L_{percept}$ in Eq. 9
Dense Mapping Network (Sec. 3. 1)
MaskVAE (Sec. 3. 2)
Alpha Blender (Sec. 3. 2)
GAN loss L_{adv}
GAN feature loss L_{feat}
VGG loss $L_{percept}$
Objective Functions (Sec. 3. 3)
Stage – II (G_B)
M^t
Fixed parameters
M^{outer}
GT Data Pair
M^{inter}
I^{outer}
I^{inter}
α
Blend
I^{blend}
GT Data Pair
M^t
I^t
Ground Truth Data Pair
GAN loss L_{adv} in Eq. 7
GAN feature loss L_{feat} in Eq. 8
VGG loss $L_{percept}$ in Eq. 9

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.

Mask VAE

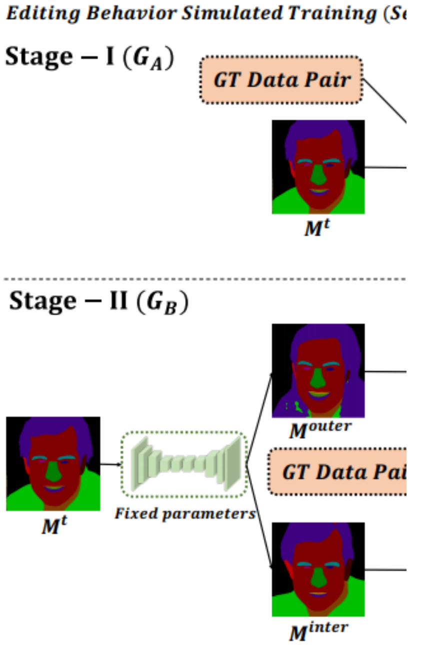


Figure 2: Overall training pipeline. Editi
Mapping Network and MaskVAE, we iterativ

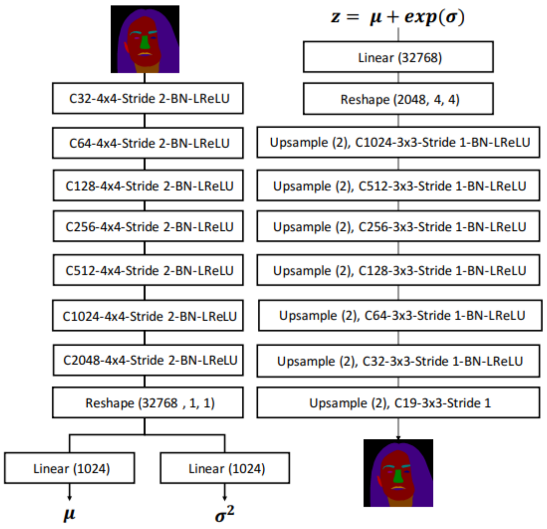

$z = \mu + exp(\sigma)$
C32-4x4-Stride 2-BN-LReLU
C64-4x4-Stride 2-BN-LReLU
C128-4x4-Stride 2-BN-LReLU
C256-4x4-Stride 2-BN-LReLU
C512-4x4-Stride 2-BN-LReLU
C1024-4x4-Stride 2-BN-LReLU
C2048-4x4-Stride 2-BN-LReLU
Reshape (32768 , 1, 1)
Linear (1024)
Linear (1024)
μ
σ^2
Linear (32768)
Reshape (2048, 4, 4)
Upsample (2), C1024-3x3-Stride 1-BN-LReLU
Upsample (2), C512-3x3-Stride 1-BN-LReLU
Upsample (2), C256-3x3-Stride 1-BN-LReLU
Upsample (2), C128-3x3-Stride 1-BN-LReLU
Upsample (2), C64-3x3-Stride 1-BN-LReLU
Upsample (2), C32-3x3-Stride 1-BN-LReLU
Upsample (2), C19-3x3-Stride 1

Figure 10: Architecture of MaskVAE.

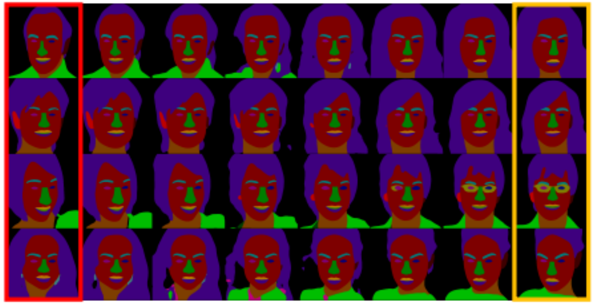
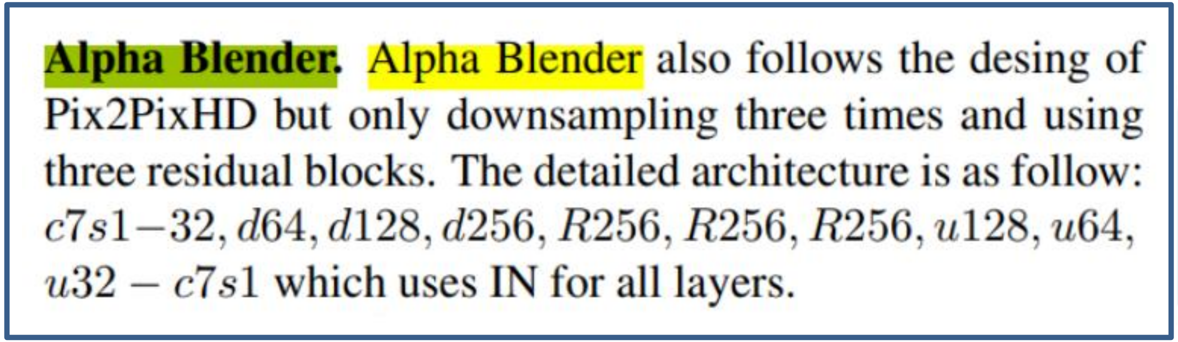


Figure 4: Samples of linear interpolation between two masks (between the red block and the orange block). MaskVAE can perform smooth transition on masks.

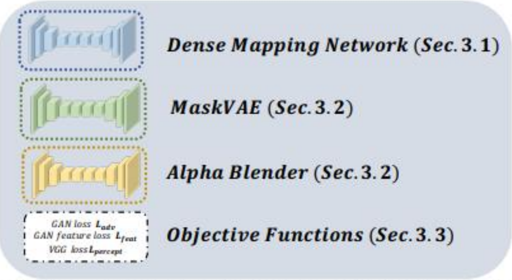
$$\pm \frac{z^{ref} - z^{t}}{\lambda_{inter}}$$

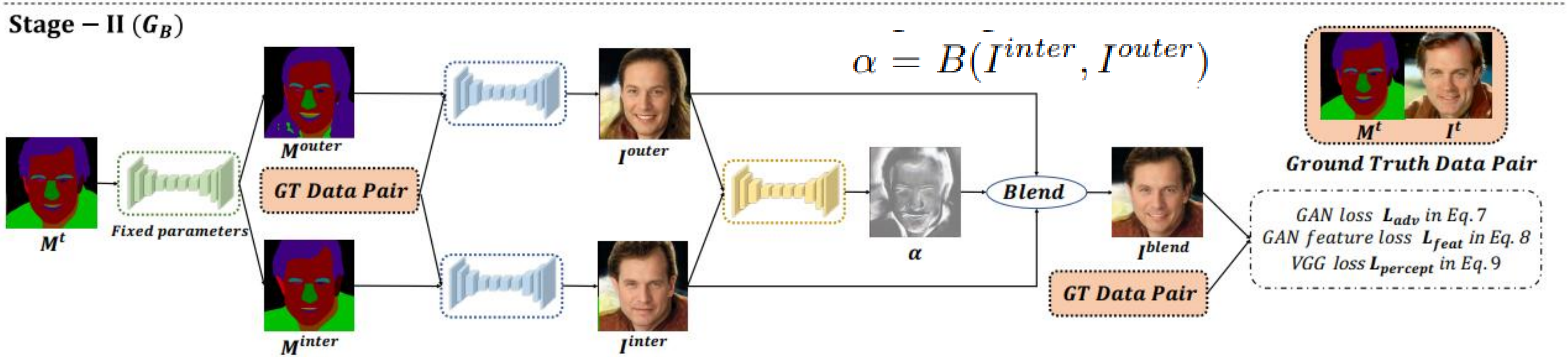
latent representation of a random selected mask M^{ref}

Alpha Blender



Alpha Blender. Alpha Blender also follows the desing of Pix2PixHD but only downsampling three times and using three residual blocks. The detailed architecture is as follow: $c7s1{-}32, d64, d128, d256, R256, R256, R256, u128, u64, u32 - c7s1$ which uses IN for all layers.


Dense Mapping Network (Sec. 3.1)
MaskVAE (Sec. 3.2)
Alpha Blender (Sec. 3.2)
GAN loss L_{adv}
GAN feature loss L_{feat}
VGG loss $L_{percept}$
Objective Functions (Sec. 3.3)



$$\alpha = B(I^{inter}, I^{outer})$$

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.

$$I^{blend} = \alpha \times I^{inter} + (1 - \alpha) \times I^{outer}$$

Training Rule

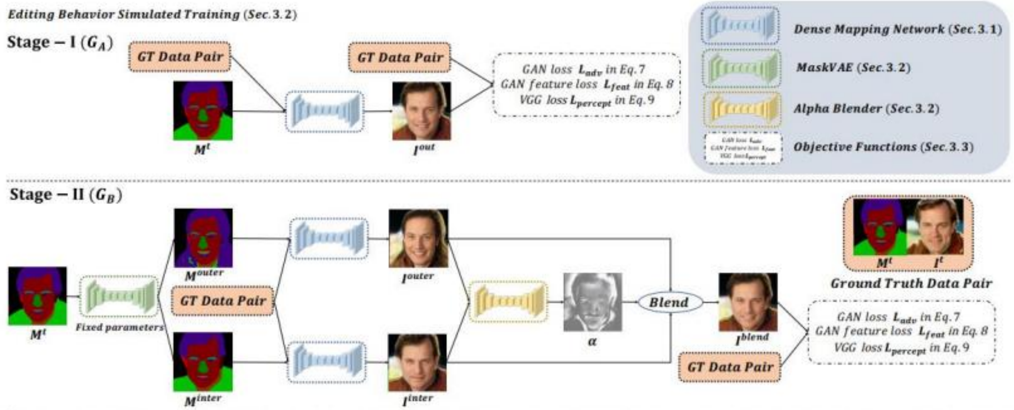
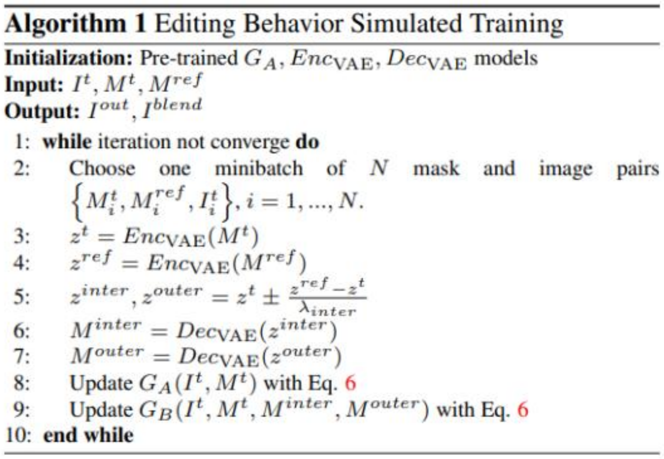

Editing Behavior Simulated Training (Sec.3.2)
Stage – I (G_A)
GT Data Pair
M^t
GT Data Pair
I^{out}
GAN loss L_{adv} in Eq.7
GAN feature loss L_{feat} in Eq.8
VGG loss $L_{percept}$ in Eq.9
Dense Mapping Network (Sec.3.1)
MaskVAE (Sec.3.2)
Alpha Blender (Sec.3.2)
Objective Functions (Sec.3.3)
Stage – II (G_B)
M^t
Fixed parameters
M^{outer}
GT Data Pair
M^{inter}
I^{outer}
I^{inter}
α
Blend
I^{blend}
GT Data Pair
M^t
I^t
Ground Truth Data Pair
GAN loss L_{adv} in Eq.7
GAN feature loss L_{feat} in Eq.8
VGG loss $L_{percept}$ in Eq.9

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.



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}

```
while iteration not converge do
    Choose one minibatch of $N$ mask and image pairs $\{M_i^t, M_i^{ref}, I_i^t\}, i = 1, ..., N$.
    $z^t = Enc_{\text{VAE}}(M^t)$
    $z^{ref} = Enc_{\text{VAE}}(M^{ref})$
    $z^{inter}, z^{outer} = z^t \pm \frac{z^{ref} - z^t}{\lambda_{inter}}$
    $M^{inter} = Dec_{\text{VAE}}(z^{inter})$
    $M^{outer} = Dec_{\text{VAE}}(z^{outer})$
    Update $G_A(I^t, M^t)$ with Eq. 6
    Update $G_B(I^t, M^t, M^{inter}, M^{outer})$ with Eq. 6
end while
```

$$\mathcal{L}_{MaskVAE} = \mathcal{L}_{reconstruct} + \lambda_{KL}\mathcal{L}_{KL}$$

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

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

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

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

CelebAMask-HQ Dataset

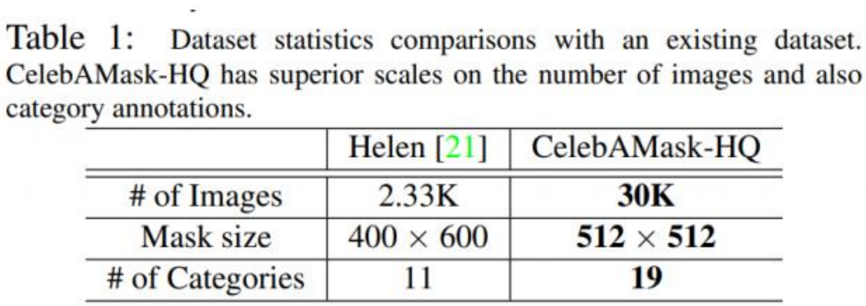
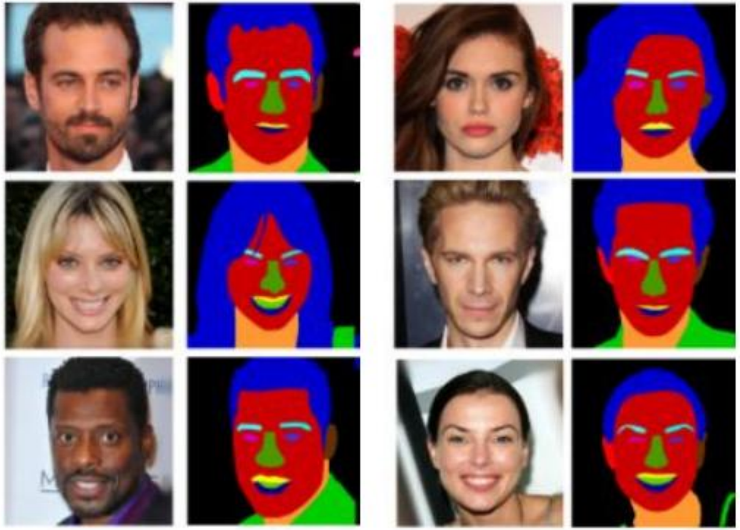
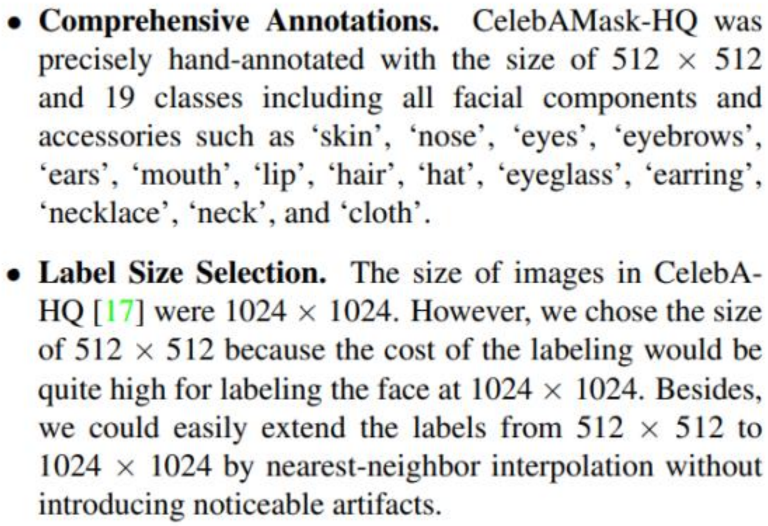


Table 1: Dataset statistics comparisons with an existing dataset. CelebAMask-HQ has superior scales on the number of images and also category annotations.

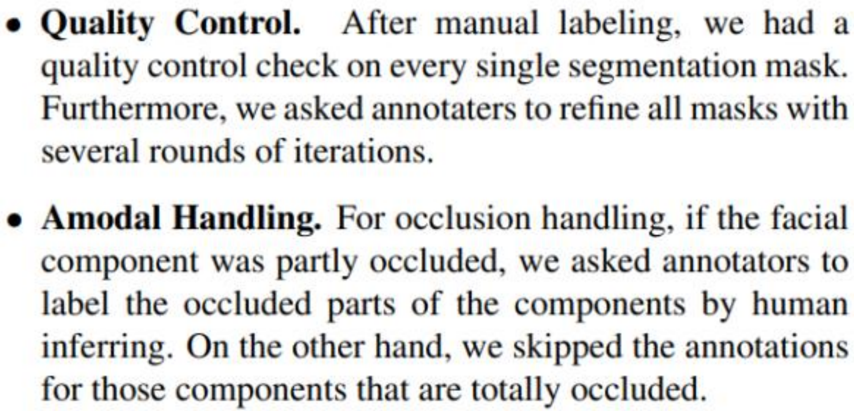
	Helen [21]	CelebAMask-HQ
# of Images	2.33K	**30K**
Mask size	400 × 600	**512 × 512**
# of Categories	11	**19**





- **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', 'earring', 'necklace', 'neck', and 'cloth'.

- **Label Size Selection.** The size of images in CelebA-HQ [17] were 1024 × 1024. However, we chose the size of 512 × 512 because the cost of the labeling would be quite high for labeling the face at 1024 × 1024. Besides, we could easily extend the labels from 512 × 512 to 1024 × 1024 by nearest-neighbor interpolation without introducing noticeable artifacts.



- **Quality Control.** After manual labeling, we had a quality control check on every single segmentation mask. Furthermore, we asked annotaters to refine all masks with several rounds of iterations.

- **Amodal Handling.** For occlusion handling, if the facial component was partly occluded, we asked annotators to label the occluded parts of the components by human inferring. On the other hand, we skipped the annotations for those components that are totally occluded.

Result1

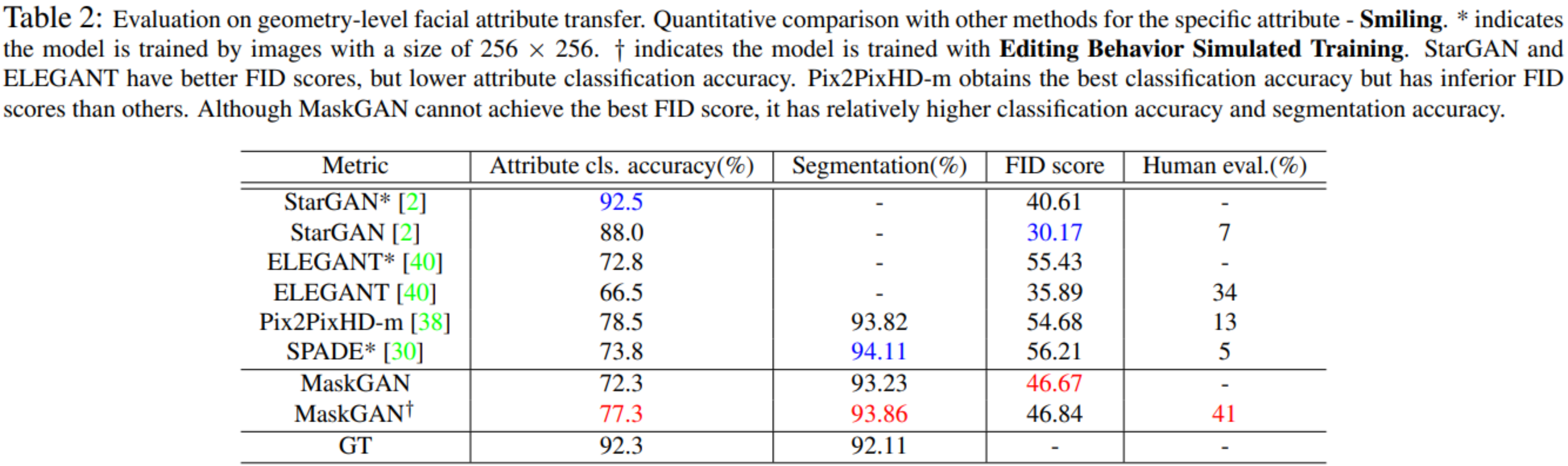


Table 2: Evaluation on geometry-level facial attribute transfer. Quantitative comparison with other methods for the specific attribute - **Smiling**. * indicates the model is trained by images with a size of 256 × 256. † indicates the model is trained with **Editing Behavior Simulated Training**. StarGAN and ELEGANT have better FID scores, but lower attribute classification accuracy. Pix2PixHD-m obtains 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.

Metric	Attribute cls. accuracy(%)	Segmentation(%)	FID score	Human eval.(%)
StarGAN* [2]	92.5	-	40.61	-
StarGAN [2]	88.0	-	30.17	7
ELEGANT* [40]	72.8	-	55.43	-
ELEGANT [40]	66.5	-	35.89	34
Pix2PixHD-m [38]	78.5	93.82	54.68	13
SPADE* [30]	73.8	94.11	56.21	5
MaskGAN	72.3	93.23	46.67	-
MaskGAN†	77.3	93.86	46.84	41
GT	92.3	92.11	-	-

Result2

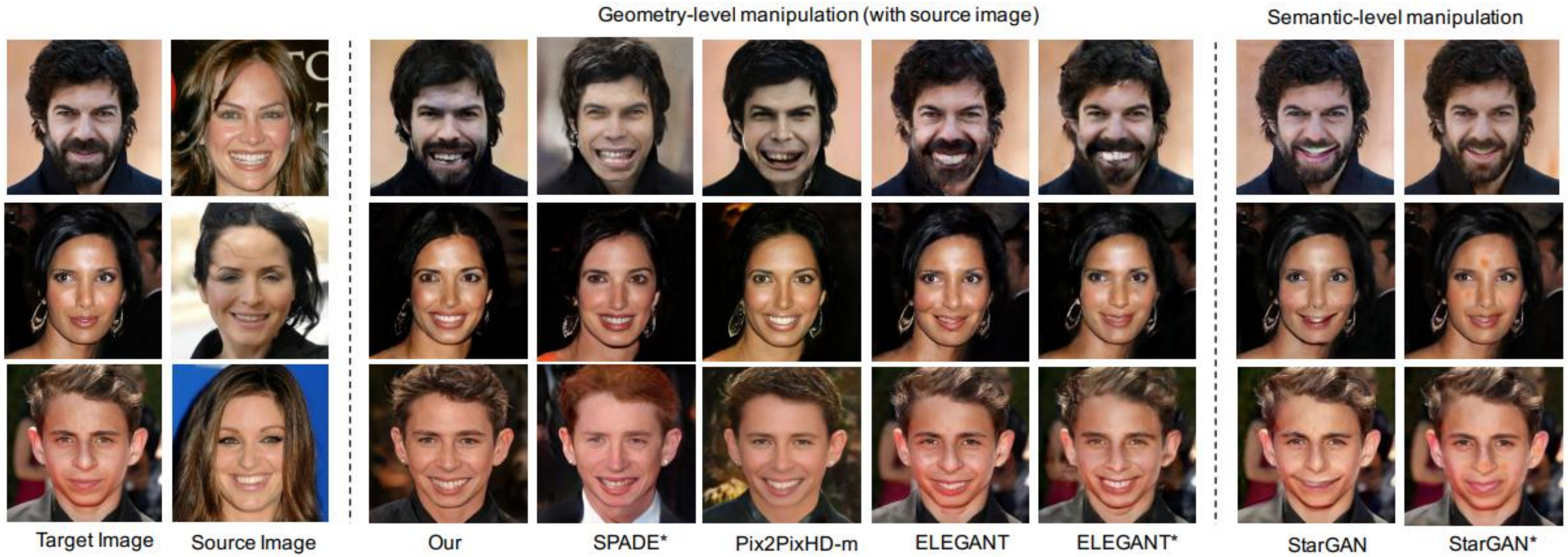

Geometry-level manipulation (with source image)
Semantic-level manipulation
Target Image
Source Image
Our
SPADE*
Pix2PixHD-m
ELEGANT
ELEGANT*
StarGAN
StarGAN*

Figure 7: Visual comparison with other methods for a specific attribute: **Smiling** on facial attribute transfer. * means the model is trained by images with a size of 256 × 256. The first two columns are target and source pairs. The middle five columns show the results of geometry-level manipulation (our MaskGAN, SPADE [30], Pix2PixHD-m [38], and ELEGANT [40]) which utilize source images as exemplars. The last two columns show the results based on semantic-level manipulation (e.g. StarGAN [2]). StarGAN fails in 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. SPADE has poor attribute keeping ability. Our MaskGAN has plausible visual quality and relatively better geometry-level smiling transferring ability.

Result3

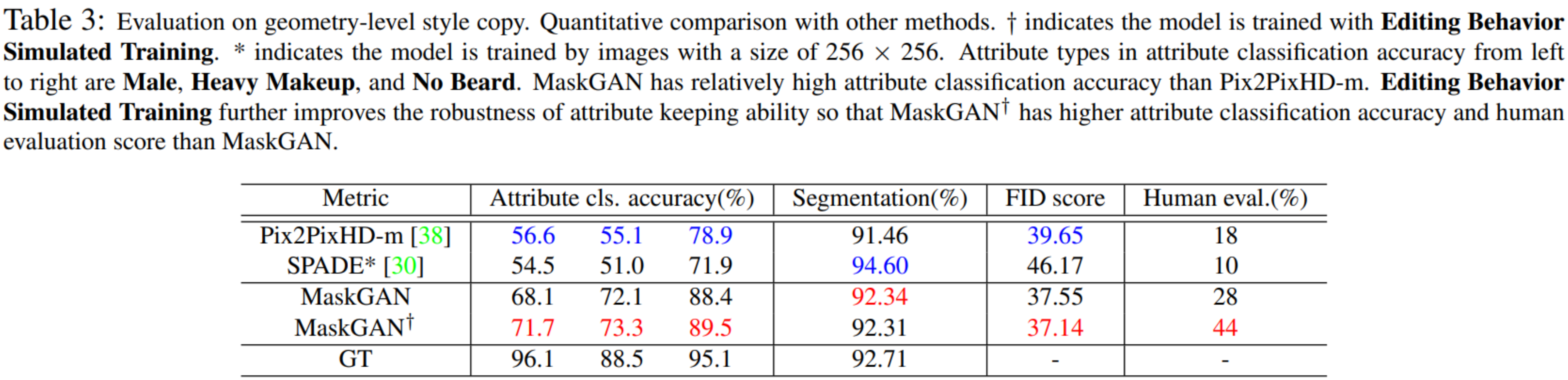


Table 3: Evaluation on geometry-level style copy. Quantitative comparison with other methods. † indicates the model is trained with **Editing Behavior Simulated Training**. * indicates the model is trained by images with a size of 256 × 256. Attribute types in attribute classification accuracy from left 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.

Metric	Attribute cls. accuracy(%)			Segmentation(%)	FID score	Human eval.(%)
Pix2PixHD-m [38]	56.6	55.1	78.9	91.46	39.65	18
SPADE* [30]	54.5	51.0	71.9	94.60	46.17	10
MaskGAN	68.1	72.1	88.4	92.34	37.55	28
MaskGAN†	71.7	73.3	89.5	92.31	37.14	44
GT	96.1	88.5	95.1	92.71	-	-

Result4

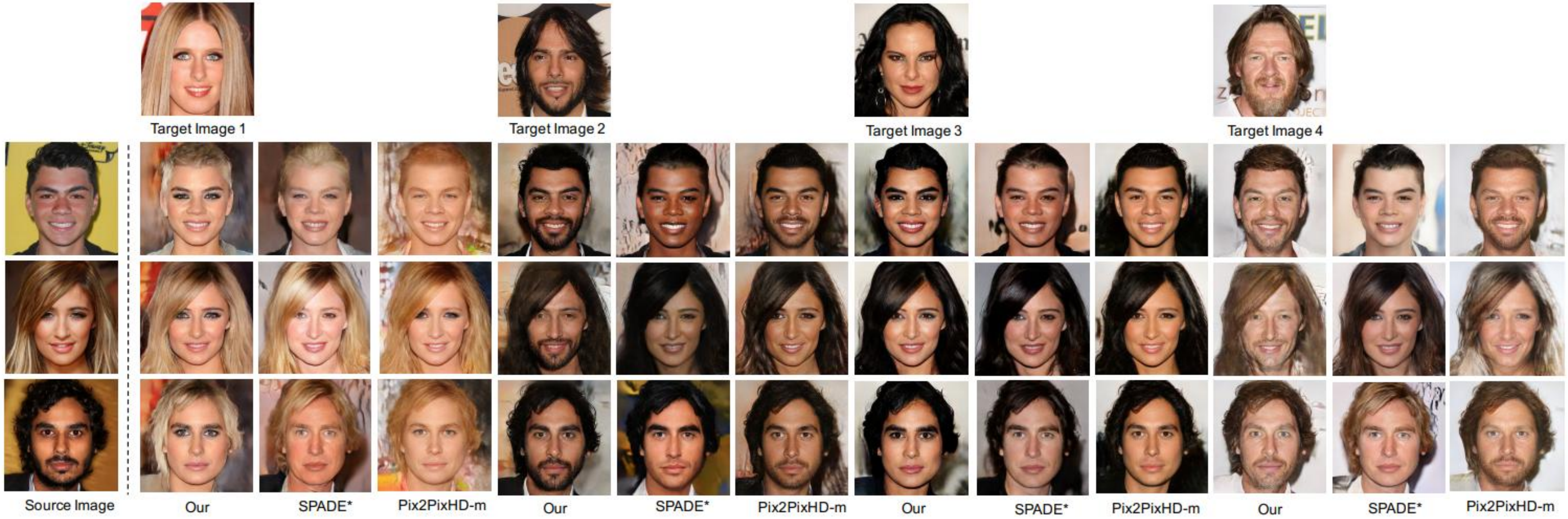
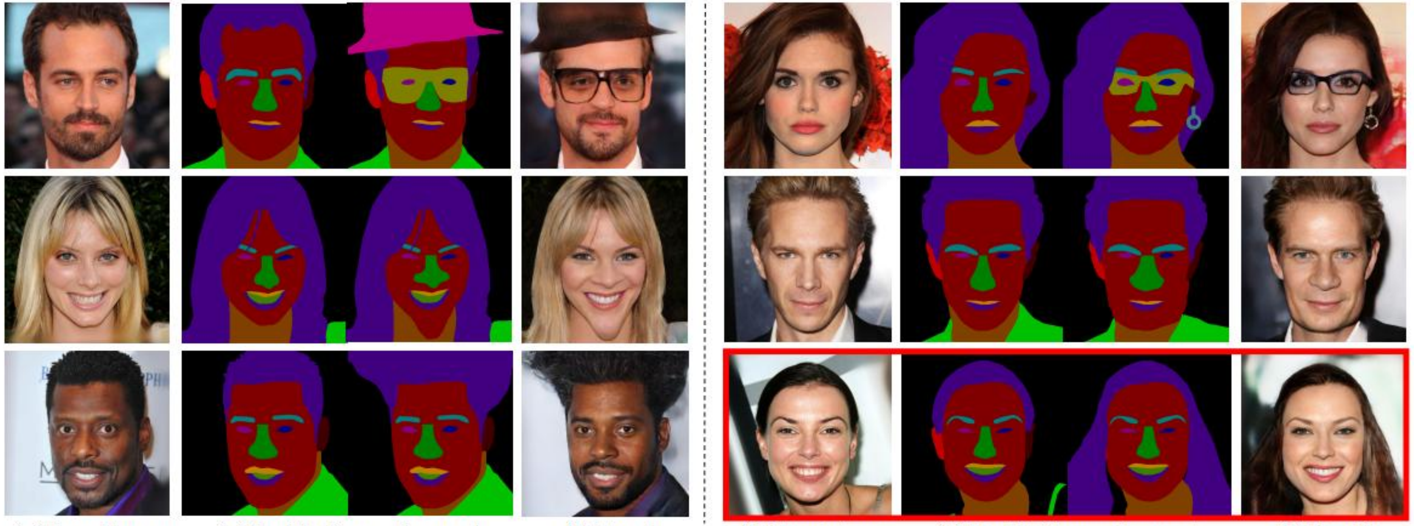


Figure 8: Visual comparison with other methods on style copy. * indicates the model is trained by images with a size of 256 × 256. All the columns show the results of the proposed method, SPADE [30] and Pix2PixHD-m [38] for four different target images. MaskGAN shows a better ability to transfer style like makeup and gender than SPADE and Pix2PixHD-m. SPADE gets better accuracy on segmentation results.

Result4



(a) Target Image (b) Modification on the mask (c) Result (a) Target Image (b) Modification on the mask (c) Result

Figure 9: Visual results of interactive face editing. The first row shows examples of adding accessories like eyeglasses. The second row shows examples of editing the shape of face and nose. The third row shows examples of adding hair. The red block shows a fail case where the strength of hair color decreases when adding hair to a short hair woman.

Reference

- https://arxiv.org/abs/1907.11922