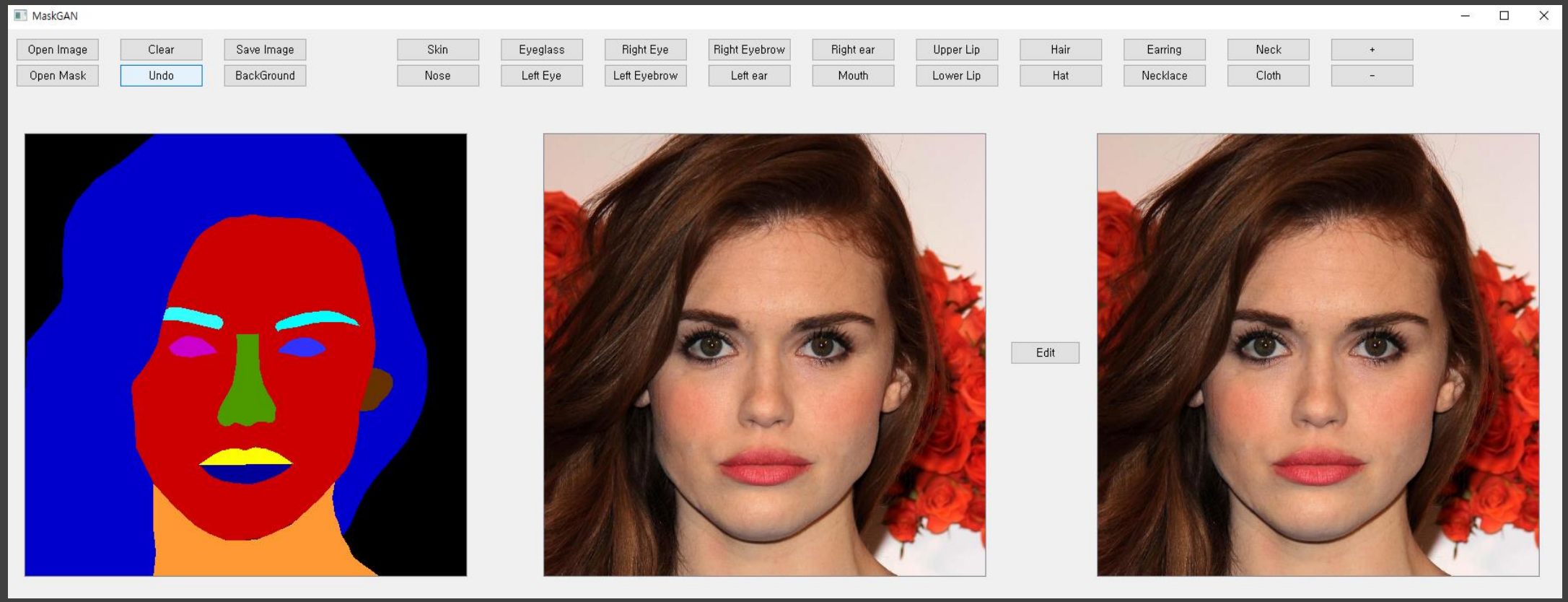
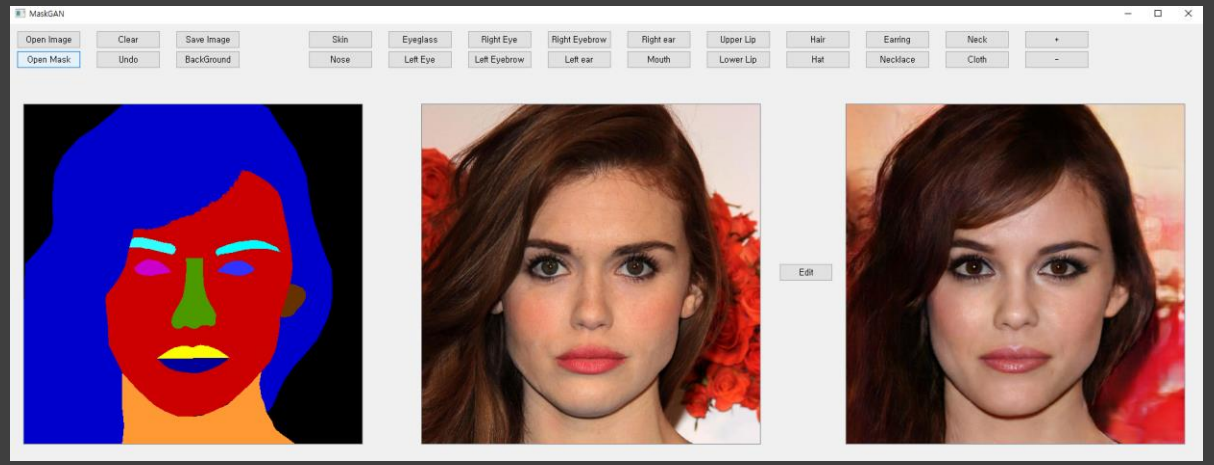
LEE, Cheng-Han, et al. Maskgan: Towards diverse and interactive facial image manipulation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020. p. 5549-5558.

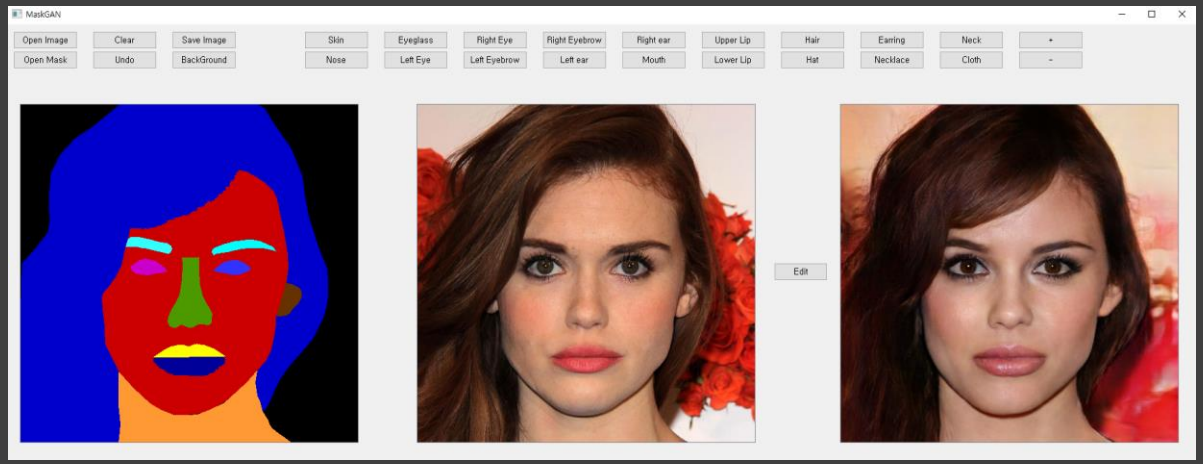
Original

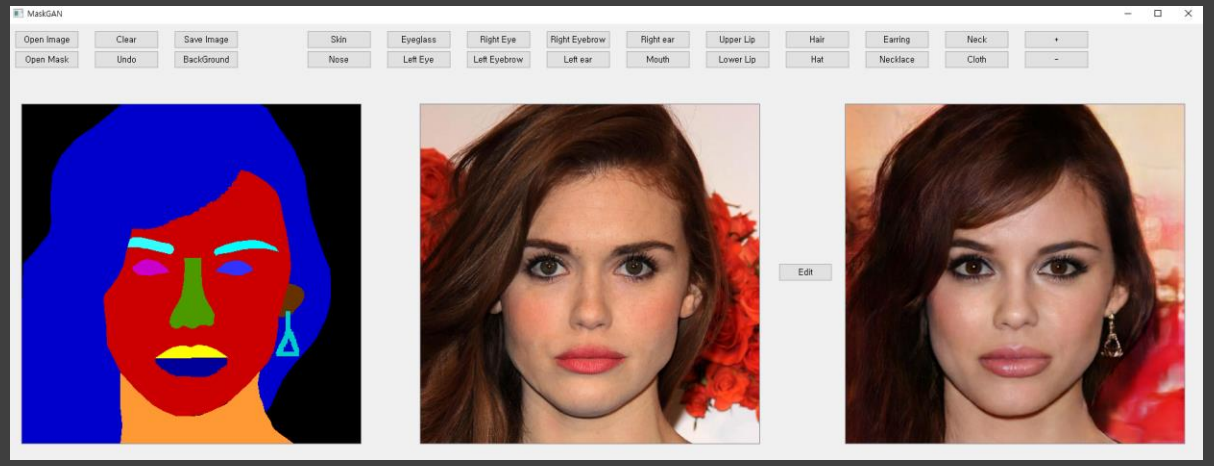

MaskGAN
Open Image
Clear
Save Image
Skin
Eyeglass
Right Eye
Right Eyebrow
Right ear
Upper Lip
Hair
Earring
Neck
+
Open Mask
Undo
BackGround
Nose
Left Eye
Left Eyebrow
Left ear
Mouth
Lower Lip
Hat
Necklace
Cloth
-
Edit



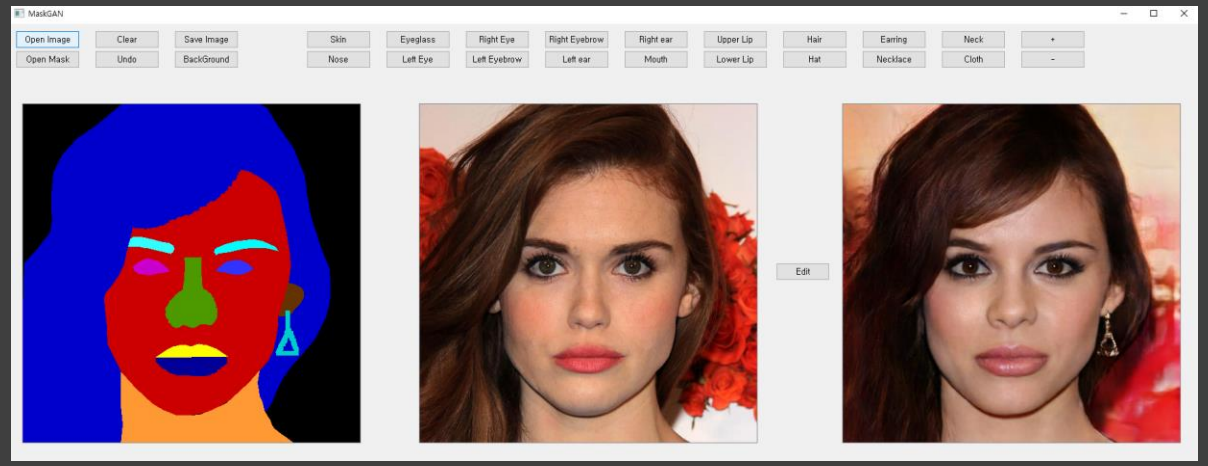
Hair

Lower Lip



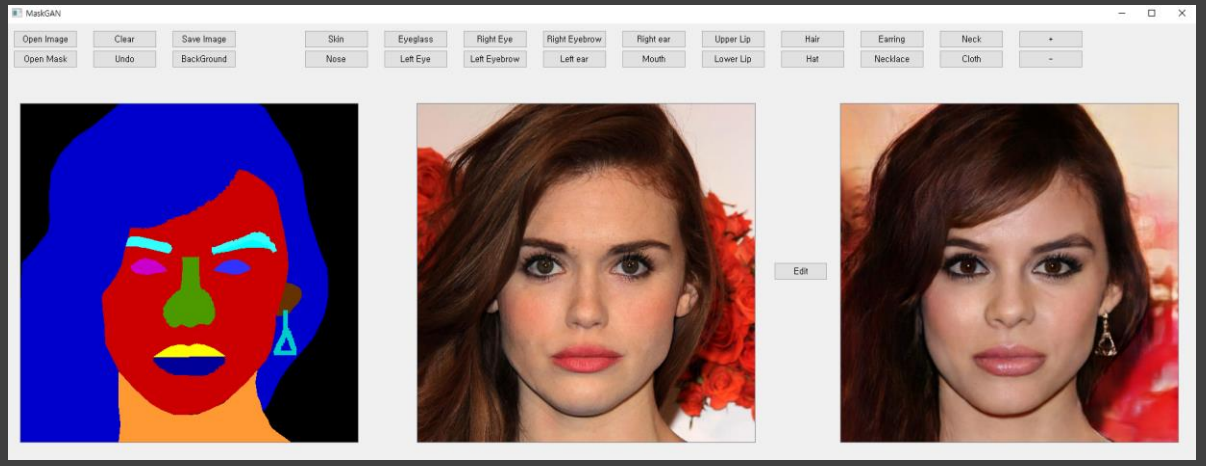


Ear ring



Nose

Right Eye Brow



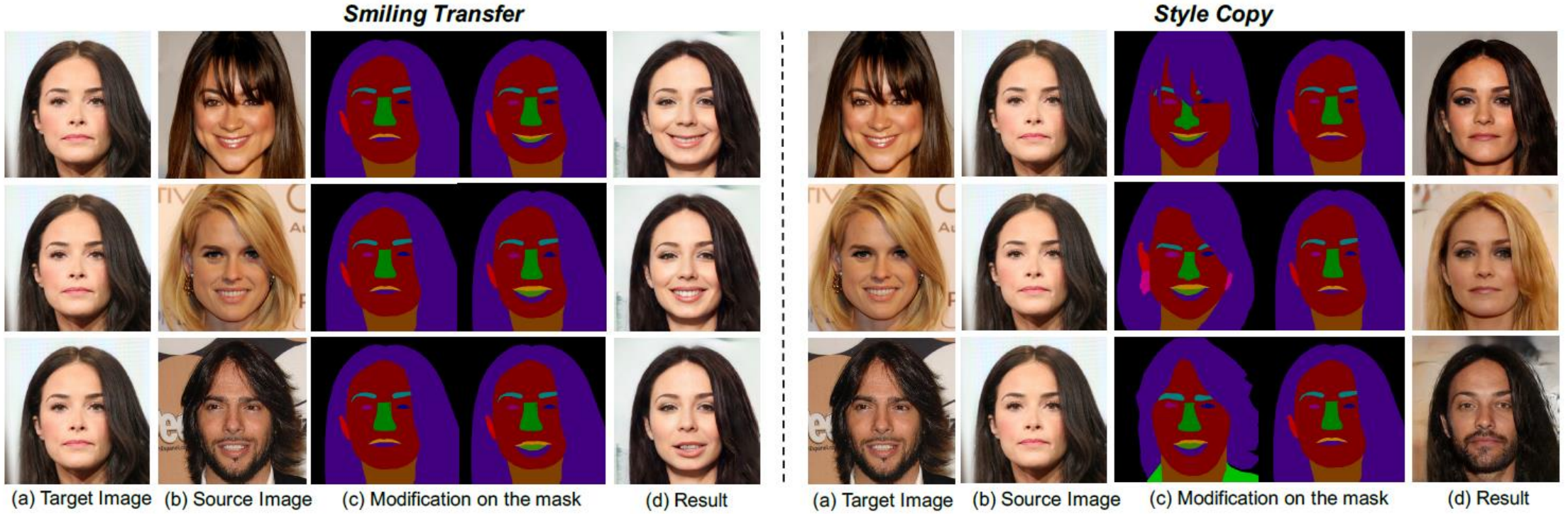


Figure 1: Given a target image (a), users are allowed to modify masks of the target images in (c) according to the source images (b) so that we can obtain manipulation results (d). The left shows illustrative examples from "neutral" to "smiling", while the right shows style copy such as makeup, hair,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

1. Introduction

- Facial image manipulation → Applications 에 사용!
  - Automatic facial expressions and styles transfer

- 두 가지 종류의 Face image manipulation 가 존재.
  - Semantic-level manipulation
  - Geometry-level manipulation
- 그러나, 이 방법들은…
  - Pre-defined face attributes set 에서 작동한다.
  - 사용자가 얼굴 이미지를 실시간으로 조작할 수 없다.

그래서 MaskGAN 을 제안하겠다!

- MaskGAN 의 Key Insight
  - Semantic masks 는 유연한 manipulation 을 하기 위한 적절한 intermediate representation 이다.

- MaskGAN 의 장점
  - MaskGAN 은 face manipulation process 를 학습하여 facial components, shapes, poses 에 대한 다양한 결과를 낸다.
  - MaskGAN 은 사용자들이 Mask 를 쉽게 편집할 수 있도록 shape, location, facial component categories 를 직관적으로 표현했다.

Contribution of MaskGAN

- Dense Mapping Network (DMN)
- Editing Behavior Simulated Training (EBST)
- CelebAMask-HQ (직접 제작)

2. Related Work

- Generative Adversarial Network
- Semantic-level Face Manipulation
- Geometry-level Face Manipulation

- Generative Adversarial Network
  - GAN
    - [7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, DavidWarde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014.
  - Image-to-image translation
    - [14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1125–1134, 2017.
    - [45] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycleconsistent adversarial networks. In Proceedings of the IEEE international conference on computer vision, pages 2223–2232, 2017.
    - [24] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In NIPS, 2017.
    - [36] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In CVPR, 2018.
    - [28] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In CVPR, 2019.
  - Image inpainting
    - [23] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In Proceedings of the European Conference on Computer Vision (ECCV), pages 85–100, 2018.
    - [42] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5505–5514, 2018.
    - [43] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In Proceedings of the IEEE International Conference on Computer Vision, pages 4471–4480, 2019.
    - [15] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In Proceedings of the IEEE International Conference on Computer Vision, pages 1745–1753, 2019.
  - Virtual try-on
    - [39] Han Yang, Ruimao Zhang, Xiaobao Guo, Wei Liu, Wangmeng Zuo, and Ping Luo. Towards photo-realistic virtual try-on by adaptively generatingpreserving image content. In CVPR. Computer Vision Foundation/IEEE, 2020.
    - [9] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. Viton: An image-based virtual try-on network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7543–7552, 2018.
    - [3] Chao-Te Chou, Cheng-Han Lee, Kaipeng Zhang, Hu-Cheng Lee, and Winston H Hsu. Pivtons: Pose invariant virtual try-on shoe with conditional image completion. In Asian Conference on Computer Vision, pages 654–668. Springer, 2018.
    - [35] Bochao Wang, Huabin Zheng, Xiaodan Liang, Yimin Chen, and Liang Lin. Toward characteristic-preserving imagebased virtual try-on network. In Proceedings of the European Conference on Computer Vision (ECCV), pages 589–604, 2018.

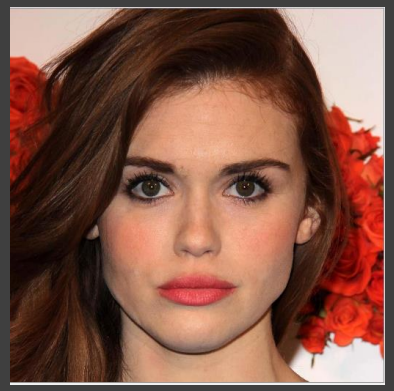
- Semantic-level Face Manipulation
  - Deep semantic-level face editing
    - [2] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In CVPR, 2018.
    - [24] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In NIPS, 2017.
    - [29] Guim Perarnau, Joost Van De Weijer, Bogdan Raducanu, and Jose M A´ lvarez. Invertible conditional gans for image editing. arXiv preprint arXiv:1611.06355, 2016.
    - [19] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, et al. Fader networks: Manipulating images by sliding attributes. In NIPS, 2017.
    - [22] Mu Li, Wangmeng Zuo, and David Zhang. Deep identity-aware transfer of facial attributes. arXiv preprint arXiv:1610.05586, 2016.
    - [21] Cheng-Han Lee, Kaipeng Zhang, Hu-Cheng Lee, Chia-Wen Cheng, andWinston Hsu. Attribute augmented convolutional neural network for face hallucination. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 721–729, 2018.
  - IcGAN
    - [29] Guim Perarnau, Joost Van De Weijer, Bogdan Raducanu, and Jose M A´ lvarez. Invertible conditional gans for image editing. arXiv preprint arXiv:1611.06355, 2016.
  - DIAT [22]
    - [22] Mu Li, Wangmeng Zuo, and David Zhang. Deep identity-aware transfer of facial attributes. arXiv preprint arXiv:1610.05586, 2016.
  - Fader Network [19]
    - [19] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, et al. Fader networks: Manipulating images by sliding attributes. In NIPS, 2017.
  - StarGAN [2]
    - [2] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In CVPR, 2018.

- Geometry-level Face Manipulation
  - Transferring facial attributes
    - [40] Raymond Yeh, Ziwei Liu, Dan B Goldman, and Aseem Agarwala. Semantic facial expression editing using autoencoded flow. arXiv preprint arXiv:1611.09961, 2016.
    - [38] Taihong Xiao, Jiapeng Hong, and Jinwen Ma. Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. arXiv preprint arXiv:1803.10562, 2018.
    - [41] Weidong Yin, Ziwei Liu, and Chen Change Loy. Instancelevel facial attributes transfer with geometry-aware flow. In AAAI, 2019.
    - [8] Shuyang Gu, Jianmin Bao, Hao Yang, Dong Chen, Fang Wen, and Lu Yuan. Mask-guided portrait editing with conditional gans. In CVPR, 2019.
  - ELEGANT
    - [38] Taihong Xiao, Jiapeng Hong, and Jinwen Ma. Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. arXiv preprint arXiv:1803.10562, 2018.
  - 3D-based face manipulation
    - [1] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. IEEE Transactions on Visualization and Computer Graphics, 20(3):413–425, 2013.
    - [27] Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, Hao Li, Richard Roberts, et al. pagan: real-time avatars using dynamic textures. ACM Trans. Graph., 37(6):258–1, 2018.
    - [6] Jiahao Geng, Tianjia Shao, Youyi Zheng, Yanlin Weng, and Kun Zhou. Warp-guided gans for single-photo facial animation. In SIGGRAPH Asia 2018 Technical Papers, page 231. ACM, 2018.

3. Our Approach

- Overall Framework
  - I^t : target image
  - M^t : semantic label mask of target image
  - M^{src} : source semantic label mask → User modified mask

I^t



M^t

M^{src}

- Training Pipeline
  - Training 시 필요한 Modules
    - Dense Mapping Network (DMN)
    - MaskVAE
    - Alpha Blender (trained by EBST)
  - Training pipeline 은 두 개의 stages 를 가진다.

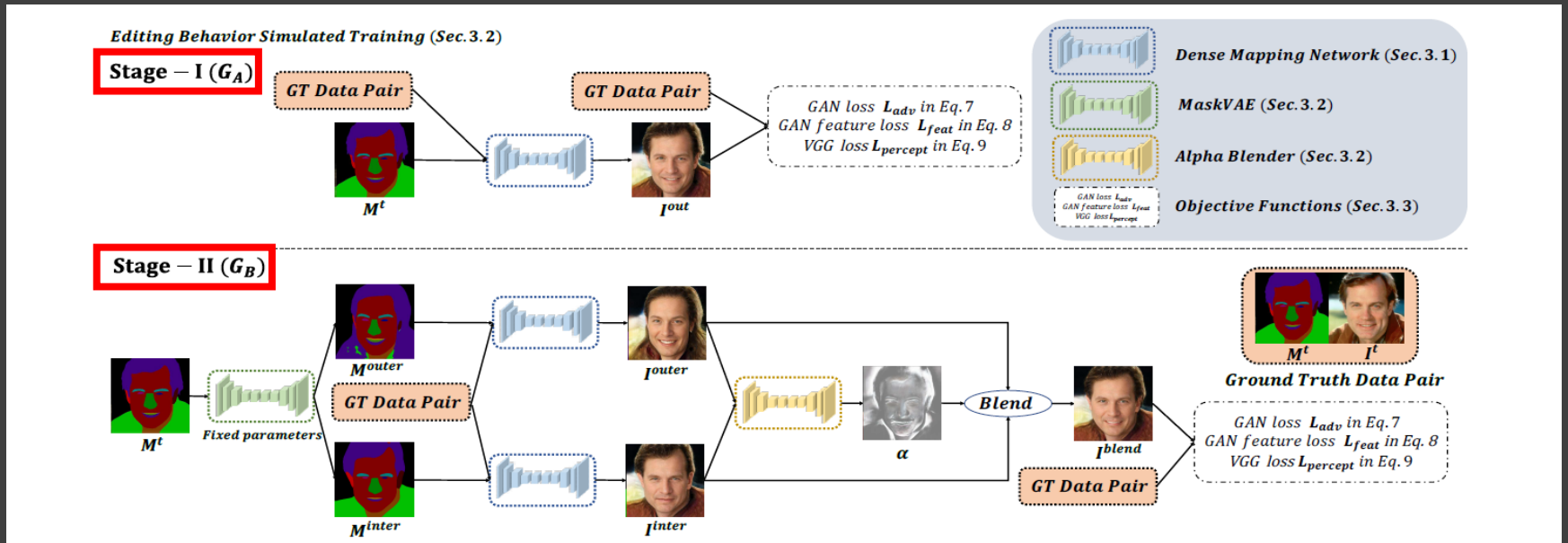
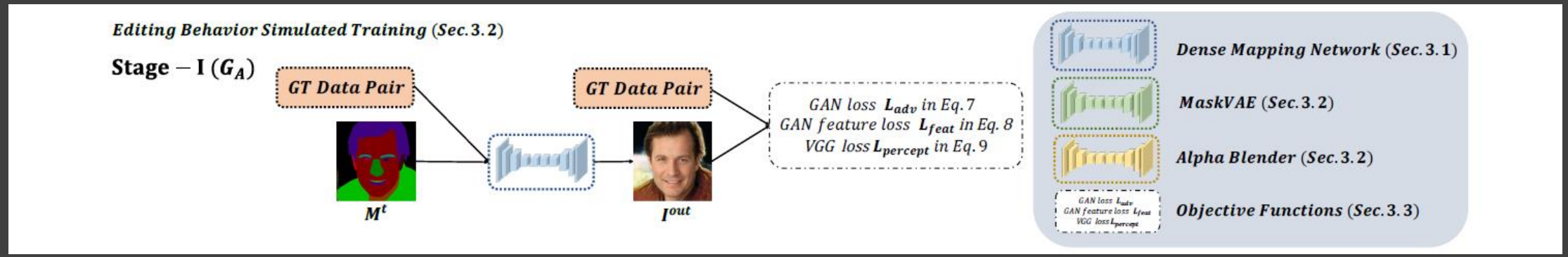


Figure 2: Overall training pipeline. Editing Behavior Simulated Training can be divided into two stage. After loading the pre-trained model of Dense Mapping Network and MaskVAE, we iteratively update these two stages until model converging.

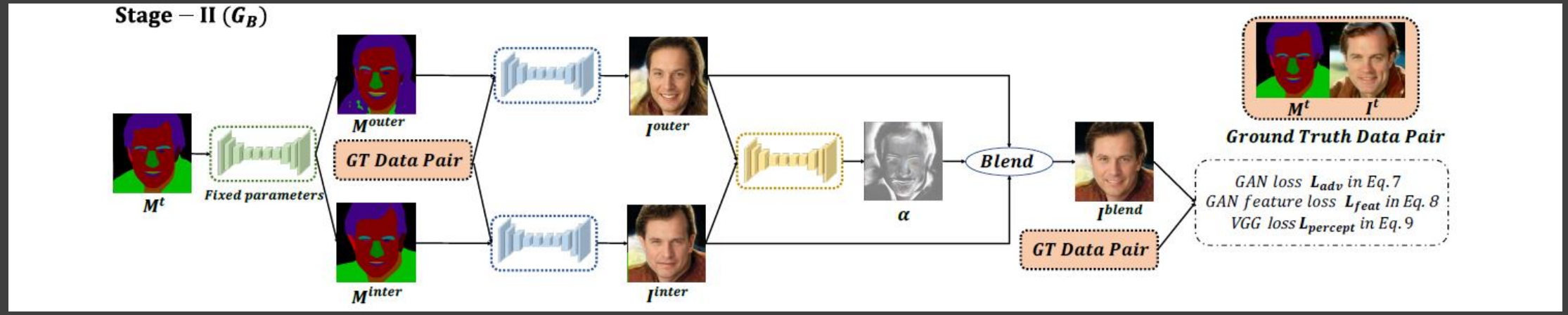
Stage 1

- M^t 와 I^t 를 가지고 DMN 을 update 한다.

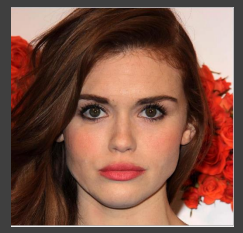


Stage 2

- MaskVAE 를 사용해서 M^t 로부터 M^{inter} 와 M^{outer} 를 만든다.
- DMN 을 통과시켜 I^{inter} 와 I^{outer} 를 만든다.
- Alpha Blender 가 이 두개의 얼굴을 $I^{blender}$ 로 합친다.



I^t M^t M^{src}



- Inference Pipeline
  - Testing 에서는 DMN 만 필요하다.
  - Image Generation Backbone 에 input 으로 M^{src} 를 넣는다.

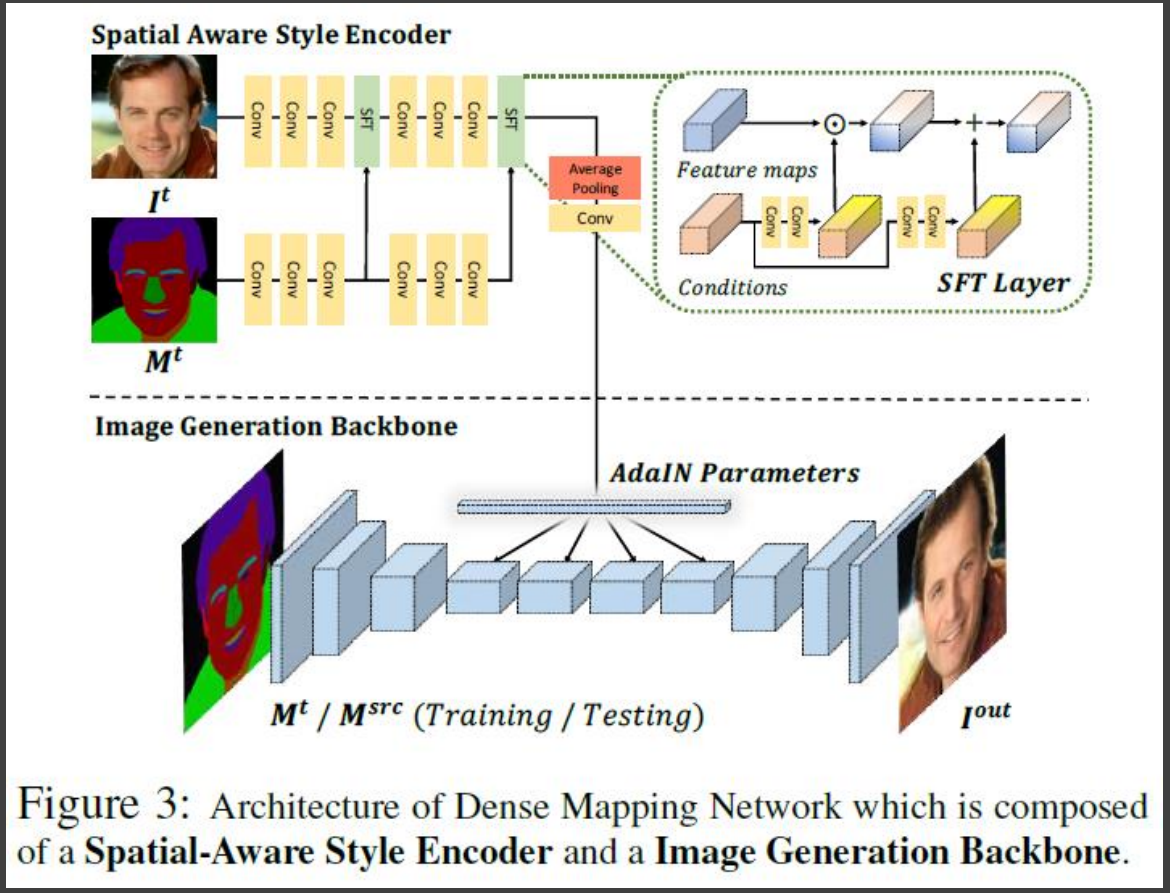


Figure 3: Architecture of Dense Mapping Network which is composed of a **Spatial-Aware Style Encoder** and a **Image Generation Backbone**.

3.1. Dense Mapping Network (DMN)

- DMN 은 (1) Spatial Aware Style Encoder 와 (2) Image Generation Backbone 으로 구성된다.

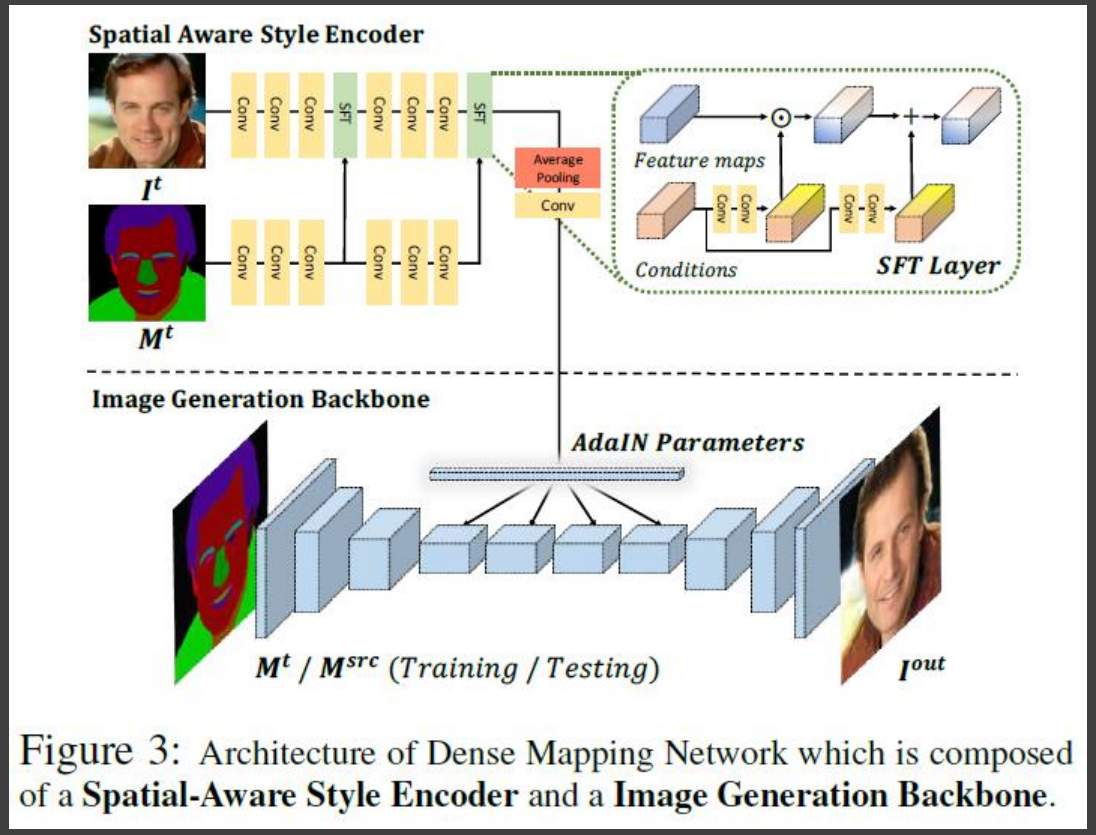
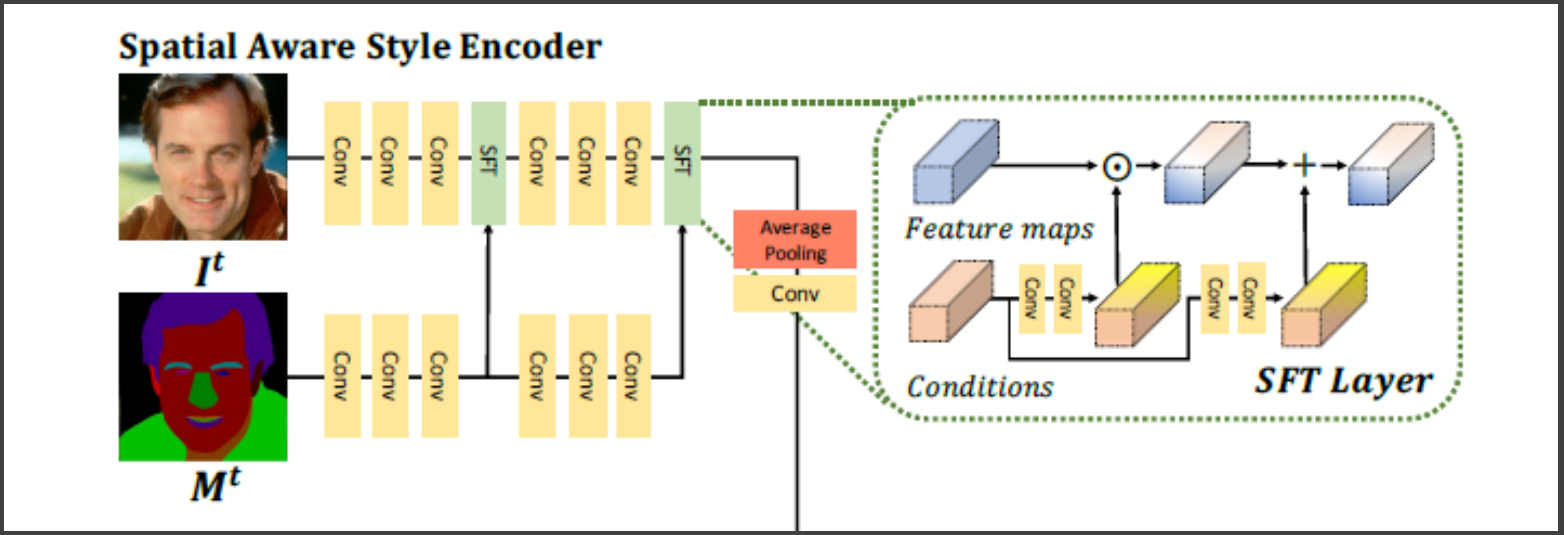


Figure 3: Architecture of Dense Mapping Network which is composed of a **Spatial-Aware Style Encoder** and a **Image Generation Backbone**.

1. Spatial-Aware Style Encoder : Enc_{style}
   - Enc_{style} 는 Pix2PixHD 를 확장 시켜서 만들었다.
   - Input 은 I^t (style information 을 제공) 와 M^t (spatial information 을 제공) 이다.
   - SFT-GAN 의 Spatial Feature Transformation(SFT) 로 I^t 와 M^t 를 합친다.

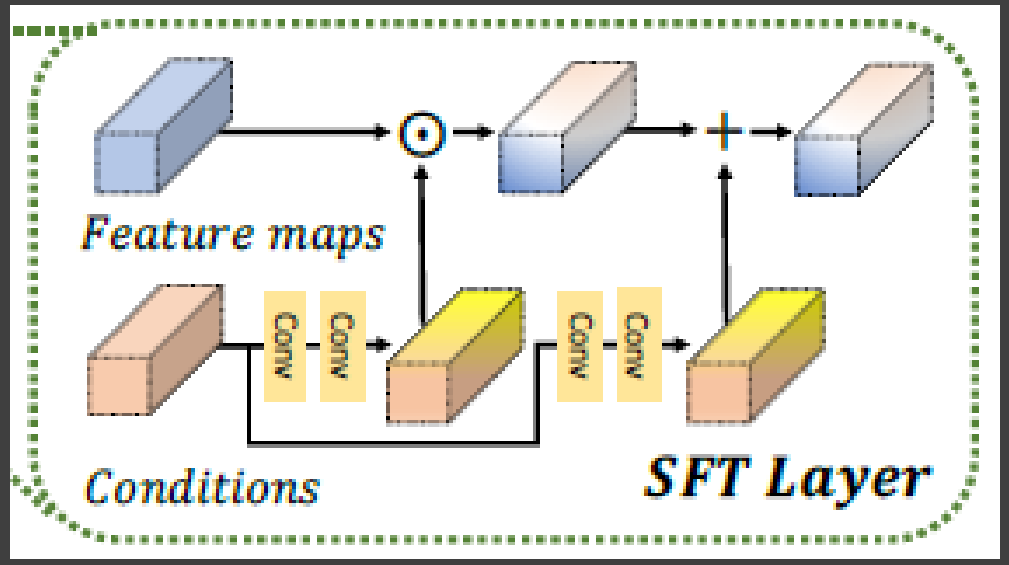


- Spatial Feature Transformation (SFT)
  - SFT layer 는 mapping function 을 사용한다.
  - Mapping function : $(\gamma, \beta) = \mathcal{M}(\Psi).$

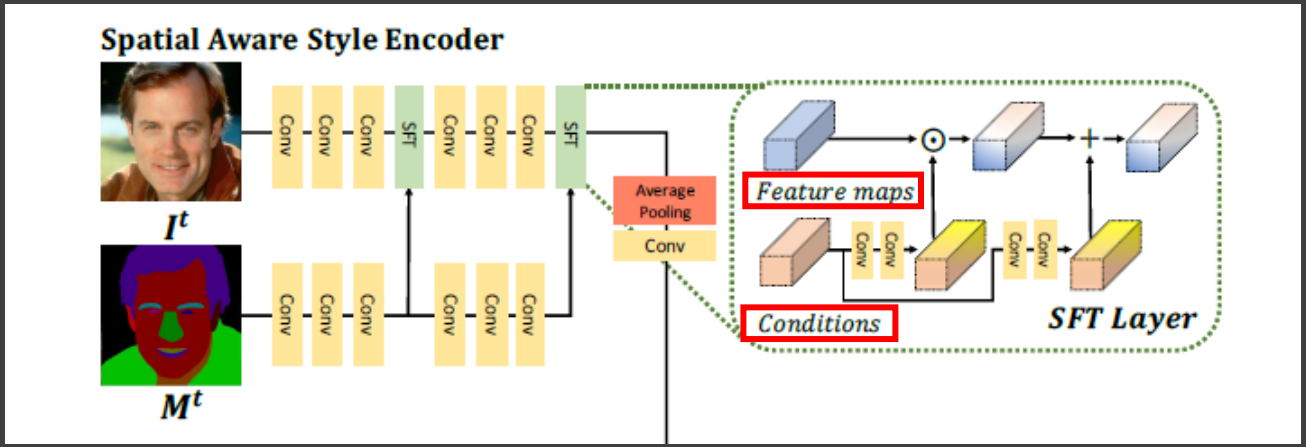
    (γ 와 β 는 affine transformation parameters, ψ 은 Prior condition)

  - γ 와 β 를 얻으면, SFT layer 가 feature map F 에서 feature-wise 와 spatial-wise modulation 을 수행한다. $SFT(F|\gamma, \beta) = \gamma \odot F + \beta$

    (F 의 차원은 γ 와 β 와 동일하고, $\odot$ 는 element-wise product 를 의미한다.)



- M^t 의 features 에서 prior-condition 인 ψ 를 얻고, I^t 에서 Feature map F 를 얻는다.



- Spatial-Aware Style Information 을 가지는 affine parameters x_i, y_i 를 얻는다.

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

2. Image Generation Backbone

- Spatial-aware style information 을 M^t 로 전달하기 위해 residual blocks z_i 을 AdaIN 에 사용한다.

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

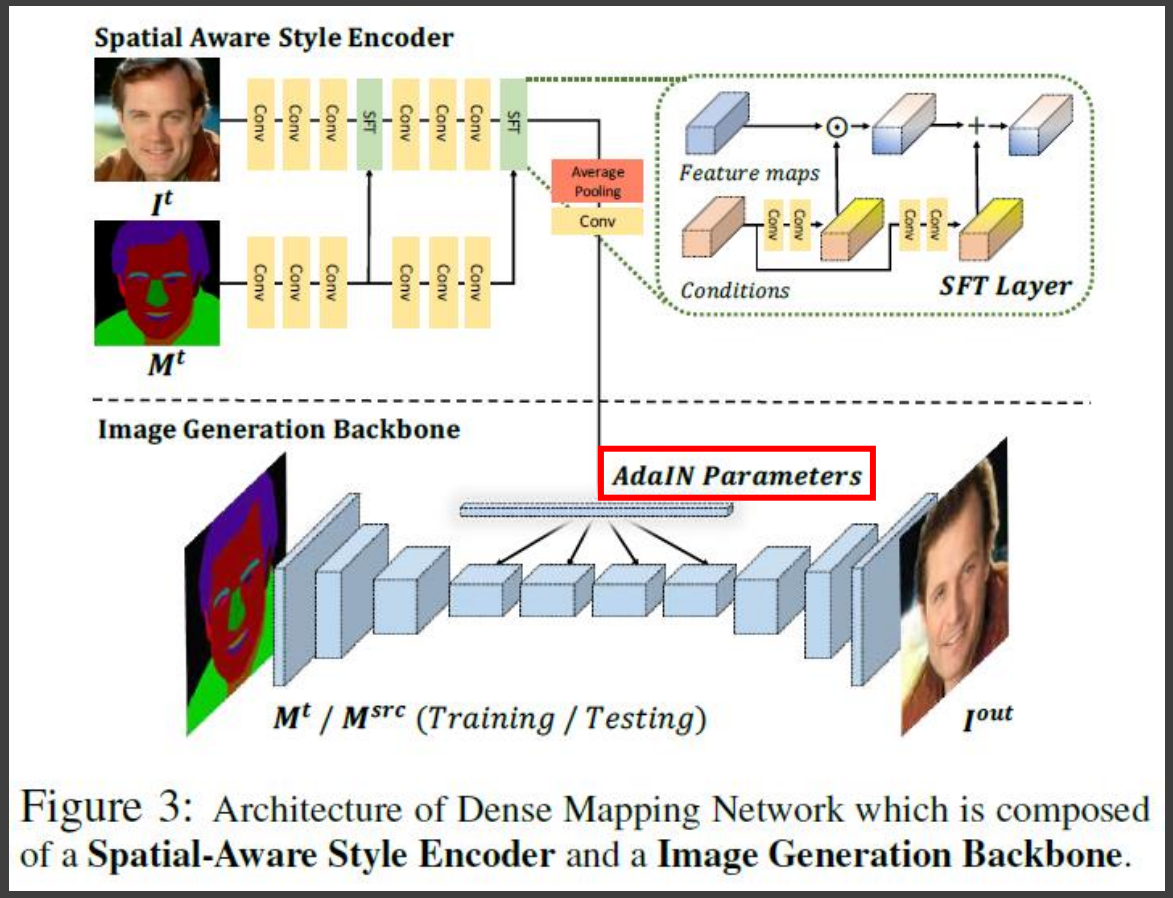


Figure 3: Architecture of Dense Mapping Network which is composed of a **Spatial-Aware Style Encoder** and a **Image Generation Backbone**.

- DMN 을 G_A(generator) 라고 할 때, $$I^{out} = G_A(Enc_{style}(I^t, M^t), M^t)).$$

I^t M^t M^{src}

- 여기 까지가 Training 에 사용되는 DMN 의 작동원리이다.

- Testing 에서 DMN 은
  - I^t 와 M^t 를 가지고 Enc_{style} 을 통해 spatial information 을 알아낸다.
  - 알아낸 spatial information 에 따라 I^t 와 M^{src} 사이에서 style mapping 을 학습한다.
  - I^t 의 style 들은 M^{src} 에 따라 바뀐다.
  - 이렇게 DMN 은 최종 manipulated face 인 I^{out} 을 만들 수 있다.

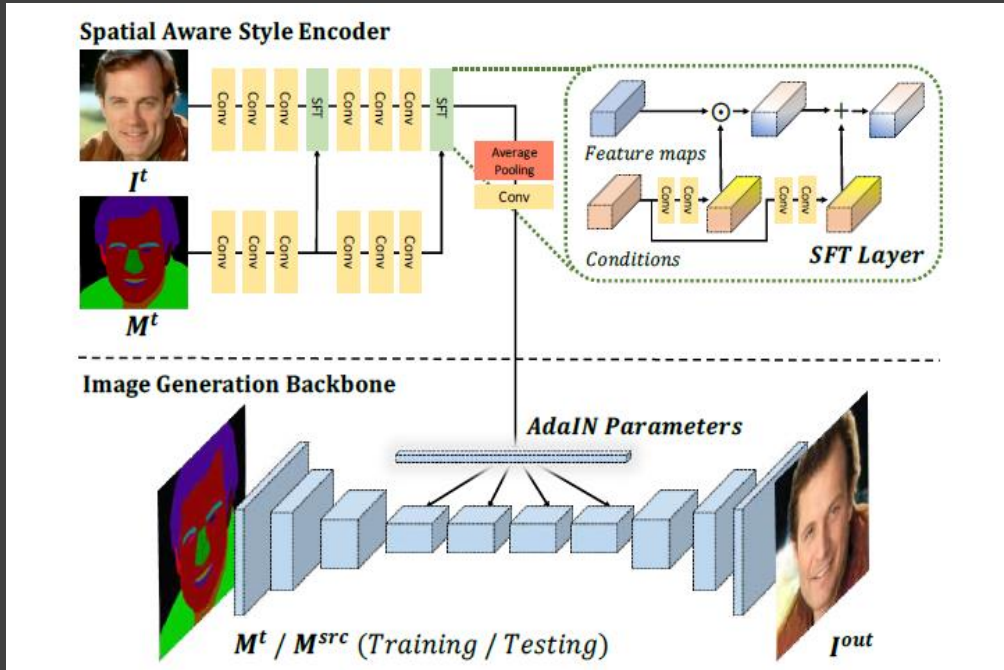


Figure 3: Architecture of Dense Mapping Network which is composed of a **Spatial-Aware Style Encoder** and a **Image Generation Backbone**.

3.2. Editing Behavior Simulated Training (EBST)

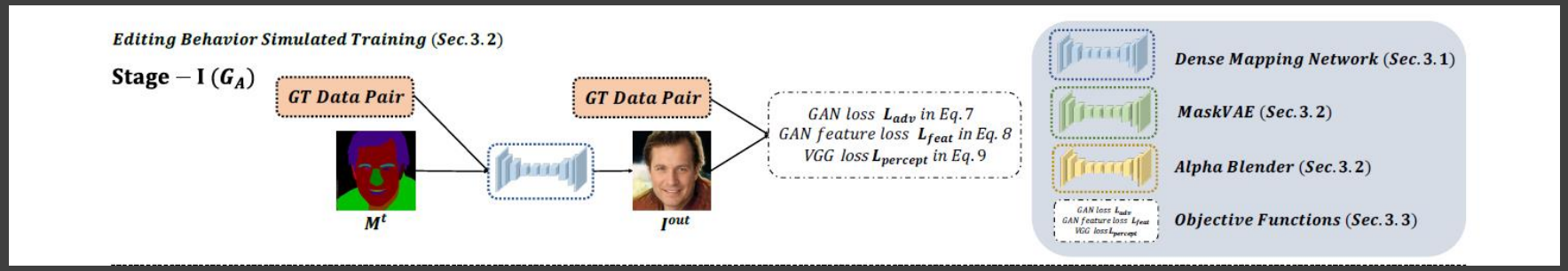
- Training time 에서, EBST 는 user 가 M^{src} 를 만드는 행동을 설계한다.
- EBST 의 구성요소
  - (Pretrained) DMN(G_A)
  - (Pretrained) MaskVAE
    - Enc_{VAE} 와 Dec_{VAE} 로 구성된다.
  - (Pretrained) Alpha Blender
    - Alpha Blender B 는 manipulation consistency 를 유지하는데 사용된다.
- G_B(generator) 를 다음과 같이 정의한다. (→ Model 들의 집합이다.)

$$G_B \equiv B(G_A(I^t, M^t, M^{inter}), G_A(I^t, M^t, M^{outer})).$$

- Training pipeline 은 두 개의 stages 로 나뉜다.

[Stage 1]

- G_A 를 Update 한다.



$$I^{out} = G_A(Enc_{style}(I^t, M^t), M^t).$$

[Stage 2]

- M^t 가 주어지면, MaskVAE 를 통해 두 개의 새로운 mask 인 M_{inter} 와 M_{outer} 를 얻는다.

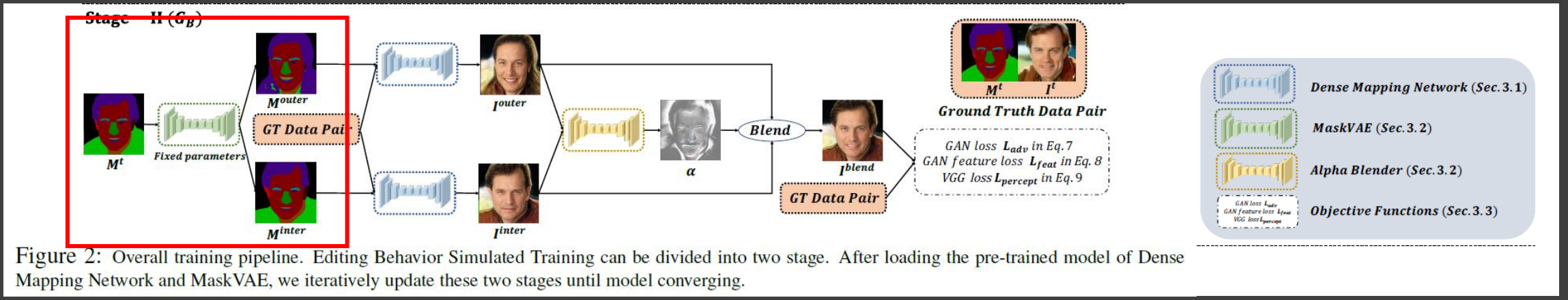


Figure 2: Overall training pipeline. Editing Behavior Simulated Training can be divided into two stage. After loading the pre-trained model of Dense Mapping Network and MaskVAE, we iteratively update these two stages until model converging.

- DMN 으로부터 두 개의 얼굴이 생성된다.

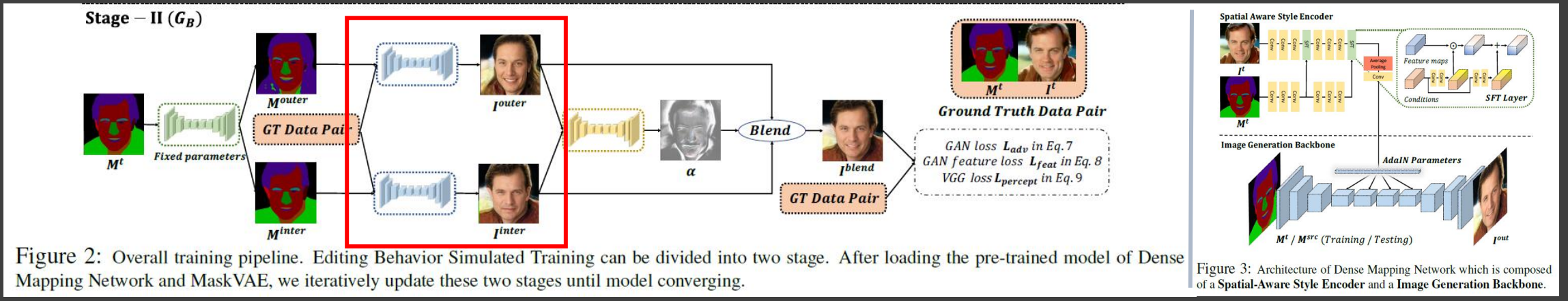


Figure 2: Overall training pipeline. Editing Behavior Simulated Training can be divided into two stage. After loading the pre-trained model of Dense Mapping Network and MaskVAE, we iteratively update these two stages until model converging.

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

- Alpha Blender 가 두 개의 image 를 blend 하는 것을 학습하고, I^{blend} 와 I^{t} 사이의 consistency 를 유지한다.

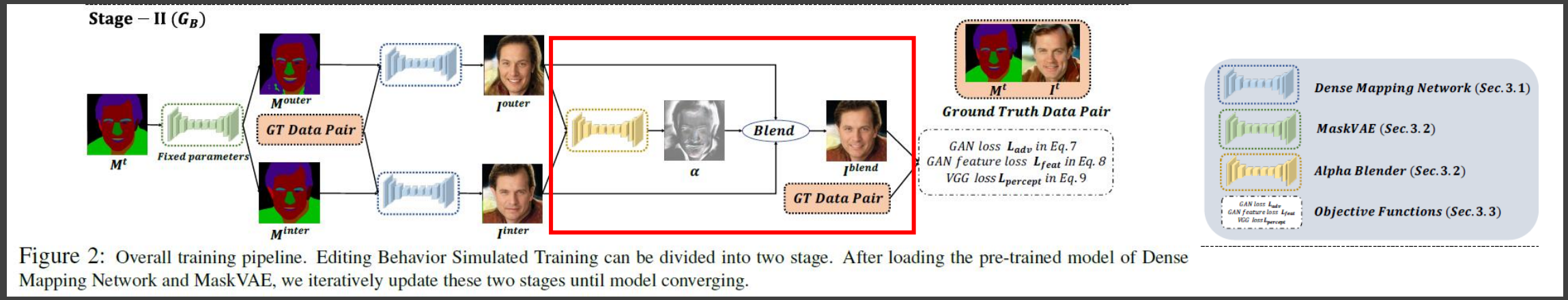
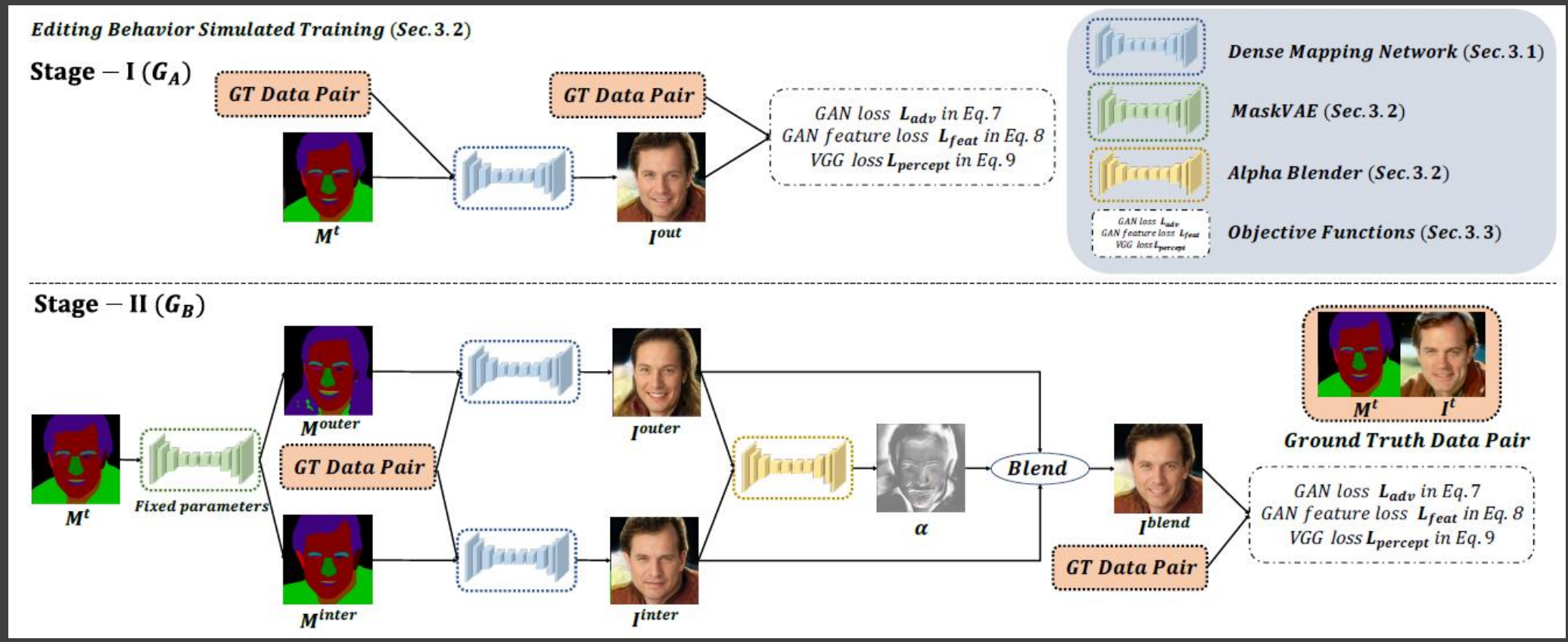


Figure 2: Overall training pipeline. Editing Behavior Simulated Training can be divided into two stage. After loading the pre-trained model of Dense Mapping Network and MaskVAE, we iteratively update these two stages until model converging.

- Model 이 만들어질 때까지 반복해서 G_A 와 G_B 를 update 한다.



$$G_B \equiv B(G_A(I^t, M^t, M^{inter}), G_A(I^t, M^t, M^{outer})).$$

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 $\left\{M_i^t, M_i^{ref}, I_i^t\right\}, i = 1, ..., N.$
  $z^t = Enc_{\text{VAE}}(M^t)$
  $z^{ref} = Enc_{\text{VAE}}(M^{ref})$
  $z^{inter}, z^{outer} = z^t \pm \frac{z^{ref} - z^t}{\lambda_{inter}}$
  $M^{inter} = Dec_{\text{VAE}}(z^{inter})$
  $M^{outer} = Dec_{\text{VAE}}(z^{outer})$
  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

- M^{ref} : random selected mask
- Z^{ref} : M^{ref} 의 latent representation
- λ_{inter} : 적절한 blending 을 위해 2.5 로 설정

3.2.1. Structural Priors by MaskVAE

- MaskVAE 의 objective function 은 두가지 파트로 구성된다.
  - $L_{resconstruct}$: pixel-wise semantic label difference 를 control 한다.
  - L_{KL} : latent space 의 smoothness 를 control 한다.

- MaskVAE 의 전체적인 loss function 은 다음과 같다.

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

- λ_{KL} 은 $1e^{-5}$ 이다. (cross validation 으로 얻었다.)

[Encoder Network]

- $Enc_{VAE}(M^t)$ 는 latent vector 의 mean μ 과 covariance σ 를 출력한다.
- 우리는 이전의 확률 분포 $P(z)$ 와 학습된 확률 분포 사이의 차이를 최소화 하기 위해 KL divergence loss 를 사용한다.

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

(여기서는 vector σ 의 $j-th$ element 를 나타낸다.)

- 그런 다음, training 단계에서 $z = \mu + r \odot exp(\sigma)$ 에 의해 latent vector z 를 만든다.
- $r \sim N(0, I)$ 은 random vector 이고, $\odot$ 은 element-wise multiplication 이다.

[Decoder Network]

- Decoder network 인 $Dec_{VAE}(\mathrm{z})$ 는 reconstruct semantic label 을 출력하고, 아래의 pixel-wise cross-entropy loss 를 계산한다.

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

- 두 개의 mask 들 사이에서 linear interpolation 한 결과를 보여준다.
- MaskVAE 는 masks 에서 smooth transition 을 수행하고, EBST 는 smooth latent 에 의존한다.

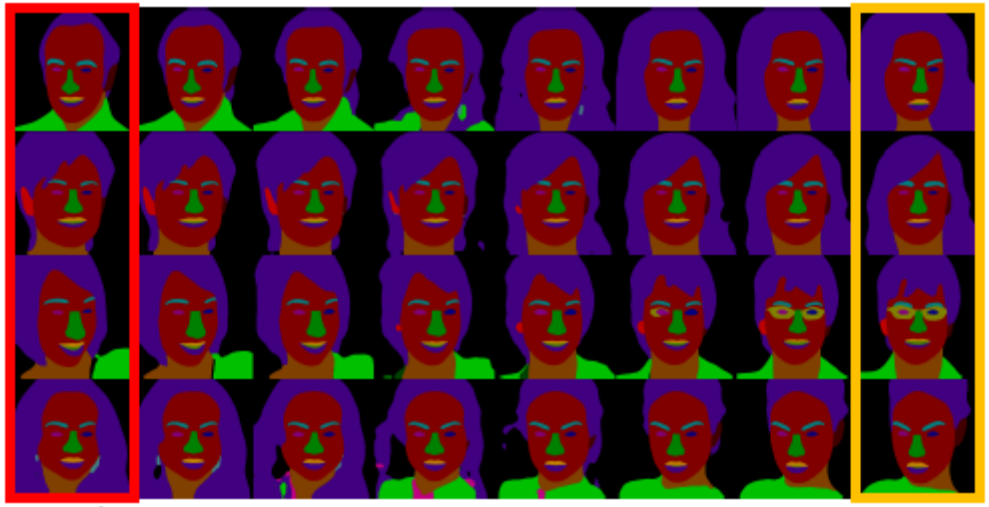


Figure 4: Samples of linear interpolation between two masks (between the red block and the orange block). MaskVAE can perform smooth transition on masks.

3.2.2. Manipulation Consistency by Alpha Blender

- I^{blend} 와 I^{t} 사이의 consistency of manipulation 을 유지하기 위해서, alpha blending 을 사용한다.
- Alpha blending 은 deep neural network 로 이미지를 구성하는데, Deep neural network 는 Alpha Blender B 에 기반을 둔다.
- B 는 두개의 이미지를 가지고 alpha blending weight α 를 학습한다.

$$I^{inter} \text{ and } I^{outer} \text{ as } \alpha = B(I^{inter}, I^{outer}).$$

- 적절한 α 를 학습한 후에, Alpha Blender 는 아래의 식에 따라 I^{inter} 와 I^{outer} 를 blend 한다. $I^{blend} = \alpha \times I^{inter} + (1 - \alpha) \times I^{outer}.$

3.3. Multi-Objective Learning

- G_A 와 G_B 의 objective function 은 세가지로 구성된다.

1. L_{adv}
   - Generated image 를 더 현실적으로 만든다.
   - M^t 에 따라 generation structure 를 수정한다.
2. L_{feat}
   - Generator 가 다양한 크기의 natural statistic 을 제공하도록 만든다.
3. $L_{percept}$
   - Content 생성을 향상시킨다. (perceptually 하게)

- Loss Function :

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

- 두 개의 다른 scale 에서 작동하기 위해 동일한 network 구조를 가진 두가지 discriminators $D_{1,2}$ 를 사용했다.
- λ_{feat} 와 $\lambda_{percept}$ 는 10으로 설정된다. (cross validation 을 통해 얻었다.)

- L_{adv} 는 conditional adversarial loss 이다.

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

- L_{feat} 은 feature matching loss 이다.
  - discriminator 로 계산한 intermediate features 를 사용해서 Real 과 Generated Image 사이의 L1 distance 를 계산한다.

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

- $L_{percept}$ 은 perceptual loss 이다.
  - fixed VGG-19 model 로 계산한 intermediate features 를 사용해서 Real 과 Generated Image 사이의 L1 distance 를 계산한다.

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

4. CelebAMask-HQ Dataset

- CelebAMask-HQ 를 제작했다.
  - CelebA 에서 30,000 개의 고화질 얼굴 이미지를 포함한 CelebA-HQ 에 따라 labeled 되었다.
  - 이 dataset 은 몇 가지 장점을 가진다.

Image

Labeled

- 장점 1) Comprehensive Annotations
  - CelebAMask-HQ 는 정확하게 hand-annotated 되었다.
  - 크기는 512 x 512
  - [skin, nose, eyes, eyebrows, ears, mouth, lip, hair, hat, eyeglass, earing, necklace, neck, cloth] 를 포함한 총 19개의 클래스를 가진다.
- 장점 2) Label Size Selection
  - CelebA-HQ 에서 이미지의 크기는 1024 x 1024 였지만, 우리는 512 x 512 크기를 선택했다. 1024 x 1024 이미지는 labeling 비용이 많이 들기 때문이다.
  - Nearest-neighbor interpolation 을 사용해서 쉽게 512 x 512 에서 1024 x 1024 로 labels 의 크기를 늘릴 수 있다.
- 장점 3) Quality Control
  - Manual labeling 후에, 우리는 모든 각 segmentation mask 에 quality control check 를 했다.
- 장점 4) Amodal Handling
  - Facial component 가 부분적으로 가려졌다면, annotators 에게 components 의 occluded parts 를 label 해달라고 부탁했다. 반면, 완전히 가려진 components 에 대해서는 annotations 를 하지 않았다.

- CelebAMask-HQ 와 Helen dataset 비교

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

	Helen [20]	CelebAMask-HQ
# of Images	2.33K	**30K**
Mask size	400 × 600	**512 × 512**
# of Categories	11	**19**

Q & A