SMILE: Semantically-guided Multi-attribute Image and Layout Editing –Supplementary Material–

Andrés Romero[1] Luc Van Gool[1,2] Radu Timofte[1]
[1]Computer Vision Lab, ETH Zürich [2]KU Leuven
{roandres, vangool, timofter}@vision.ee.ethz.ch

A. Network Architectures

In Table 1, 2 and 3, we describe our networks for the semantic manipulation stage. Similarly, Table 4, 5, and 6 shows the description for the image synthesis stage. For the latter, we bypass the description of the discriminator as it is a copy of the StyleGAN2 one. For both stages, and in order to maintain both environments as unaltered as possible, we use the same training hyperparameters (Adam Beta optimizers and learning rates) as in the corresponding baselines StarGANv2 and StyleGAN2, respectively.

B. Additional Results for Semantic Manipulation

In this section we present additional results for the semantic manipulation. We report FID, LPIPS, F1, AP, and other metrics reported in the main paper for each attribute manipulation.

In Table 7 and 8, we show quantitative evaluation for each attribute independently. Not surprisingly, Male and Female attributes are easier manipulations than hat and earrings. We also set a benchmark for the evaluation of these kind of manipulations.

Furthermore, Figure 1 depicts random and reference synthesis images.

Figure 2, 3, 4 5 6 7 and 8 depict qualitative visualizations for each attribute independently.

Part	Input → Output Shape	Layer Information
Down-sampling	$(256, 256, \mathbb{N}_{mask}) \rightarrow (256, 256, 32)$	Conv2d(dim_out=32, kernel=3, stride=1, padding=1)
	$(256, 256, 32) \rightarrow (128, 128, 64)$	Residual Block: IN, LReLU, Conv2d(64, 4, 2, 1), AvgPool2D
	$(128, 128, 64) \rightarrow (64, 64, 128)$	Residual Block: IN, LReLU, Conv2d(128, 4, 2, 1), AvgPool2D
	$(64, 64, 128) \rightarrow (32, 32, 256)$	Residual Block: IN, LReLU, Conv2d(256, 4, 2, 1), AvgPool2D
	$(32, 32, 256) \rightarrow (16, 16, 512)$	Residual Block: IN, LReLU, Conv2d(256, 4, 2, 1), AvgPool2D
	$(16, 16, 512) \rightarrow (8, 8, 512)$	Residual Block: IN, ReLU, Conv2d(256, 3, 1, 1), AvgPool2D
	$(8, 8, 512) \rightarrow (8, 8, 512)$	Residual Block: IN, LReLU, Conv2d(256, 3, 1, 1)
	$(8, 8, 512) \rightarrow (8, 8, 512)$	Residual Block: IN, LReLU, Conv2d(256, 3, 1, 1)
Up-sampling	$(8, 8, 512) \rightarrow (8, 8, 512)$	LReLU, ModulatedConv2d(256, 3, 1, 1
	$(8, 8, 512) \rightarrow (8, 8, 512)$	LReLU, ModulatedConv2d(256, 3, 1, 1)
	$(8, 8, 512) \rightarrow (16, 16, 512)$	LReLU, NearestUp, ModulatedConv2d(128, 3, 1, 1)
	$(16, 16, 256) \rightarrow (32, 32, 256)$	LReLU, NearestUp, ModulatedConv2d(128, 3, 1, 1)
	$(32, 32, 256) \rightarrow (64, 64, 128)$	LReLU, NearestUp, ModulatedConv2d(128, 3, 1, 1)
	$(64, 64, 128) \rightarrow (128, 128, 64)$	LReLU, NearestUp, ModulatedConv2d(64, 3, 1, 1)
	$(128, 128, 64) \rightarrow (256, 256, 32)$	LReLU, NearestUp, ModulatedConv2d(32, 3, 1, 1)
	$(256, 256, 32) \rightarrow (256, 256, \mathbb{N}_{mask})$	ReLU, ModulatedConv2d($\mathbb{N}_{mask}$, 3, 1, 1)

Table 1: **SMILE Semantic Manipulation Generator network architecture.** As we use the CelebA-Mask dataset [4], we set $\mathbb{N}_{mask}$ to 19.

Layer	Input $\rightarrow$ Output Shape	Layer Information
Input Layer	$(256, 256, \mathbb{N}_{mask}) \rightarrow (256, 256, 64)$	Conv2d(dim_out=64, kernel=3, stride=1, padding=1), LReLU
Hidden Layer	$(256, 256, 64) \rightarrow (128, 128, 128)$	Residual Block: LReLU, Conv2d(128, 3, 1, 1), AvgPool2D
Hidden Layer	$(128, 128, 128) \rightarrow (64, 64, 256)$	Residual Block: LReLU, Conv2d(128, 3, 1, 1), AvgPool2D
Hidden Layer	$(64, 64, 256) \rightarrow (32, 32, 512)$	Residual Block: LReLU, Conv2d(128, 3, 1, 1), AvgPool2D
Hidden Layer	$(32, 32, 512) \rightarrow (16, 16, 512)$	Residual Block: LReLU, Conv2d(256, 3, 1, 1), AvgPool2D
Hidden Layer	$(16, 16, 256) \rightarrow (8, 8, 512)$	Residual Block: LReLU, Conv2d(512, 3, 1, 1), AvgPool2D
Hidden Layer	$(8, 8, 512) \rightarrow (4, 4, 512)$	Residual Block: LReLU, Conv2d(512, 3, 1, 1), AvgPool2D
Hidden Layer	$(4, 4, 512) \rightarrow (1, 1, 512)$	LReLU, Conv2d(512, 4, 1, 0)
Output Layer	$(1, 1, 512) \rightarrow (1, 1, d \times N \times 2)$	LReLU, DS_Layer(512, d×N×2, 1, 0)

Table 2: **SMILE Semantic Manipulation Discriminator and Style Encoder network architecture.** The difference between the two networks lie in the last layer, where *DS_Layer* is a Convolution and Linear layer and *d* is the number of dimensions equal to 1 and 64 for Discriminator and Style Encoder, respectively. As we consider each domain as the presence or absence of each attribute, the number of outputs is scaled by 2. Note that as part of our method we weight the style dimensions for Male/Female, so all the remaining are in a 4:1 ratio.

Layer	Input $\rightarrow$ Output Shape	Layer Information
Shared Layers	$(16) \rightarrow (512)$	Linear(dim_out=512), ReLU
	$(512) \rightarrow (512)$	Linear(dim_out=512), ReLU
	$(512) \rightarrow (512)$	Linear(dim_out=512), ReLU
	$(512) \rightarrow (512)$	Linear(dim_out=512), ReLU
Unshared Layers	$(512) \rightarrow (512)$	Linear(dim_out=512), ReLU
	$(512) \rightarrow (512)$	Linear(dim_out=512), ReLU
	$(512) \rightarrow (512)$	Linear(dim_out=512), ReLU
	$(512) \rightarrow (WS)$	Linear(dim_out=WS)

Table 3: **SMILE Semantic Manipulation Mapping network architecture.** All attributes share the first part of the network, and each attribute has independent branch of unshared layers. *WS* stands for the Weighted Style to each attribute. For Male/Female we set 64, and 16 for other attributes. Not that presence and absence of each attribute have two independent branches in the mapping network.

Part	Input → Output Shape	Layer Information
Mask Feature Extractor	$(64, 64, \mathbb{N}_{mask}) \rightarrow (64, 64, 256)$	EqualConv2d(dim_out=32, kernel=3, stride=1, padding=1)
	$(64, 64, 256) \rightarrow (64, 64, 256)$	FusedLeakyReLU
	$(64, 64, 256) \rightarrow (32, 32, 256)$	Blur, EqualConv2d(256, 3, 2, 0), FusedLeakyReLU
	$(32, 32, 256) \rightarrow (16, 16, 256)$	Blur, EqualConv2d(256, 3, 2, 0), FusedLeakyReLU
	$(16, 16, 256) \rightarrow (8, 8, 512)$	Blur, EqualConv2d(512, 3, 2, 0), FusedLeakyReLU
Up-sampling	$(8, 8, 512) \rightarrow (8, 8, 512)$	SAC(dim_out=512, kernel=3, upsample=False)
	$(8, 8, 512) \rightarrow (8, 8, 512)$	Noise, FusedLeakyReLU
	$(8, 8, 512) \rightarrow (16, 16, 512)$	SAC(512, 3, True), Noise, FusedLeakyReLU
	$(16, 16, 512) \rightarrow (16, 16, 512)$	SAC(512, 3, False), Noise, FusedLeakyReLU
	$(16, 16, 512) \rightarrow (32, 32, 512)$	SAC(512, 3, True), Noise, FusedLeakyReLU
	$(32, 32, 512) \rightarrow (32, 32, 512)$	SAC(512, 3, False), Noise, FusedLeakyReLU
	$(32, 32, 512) \rightarrow (64, 64, 256)$	SAC(256, 3, True), Noise, FusedLeakyReLU
	$(64, 64, 256) \rightarrow (64, 64, 256)$	SAC(256, 3, False), Noise, FusedLeakyReLU
	$(64, 64, 256) \rightarrow (128, 128, 256)$	SAC(256, 3, True), Noise, FusedLeakyReLU
	$(128, 128, 128) \rightarrow (128, 128, 256)$	SAC(256, 3, False), Noise, FusedLeakyReLU
	$(128, 128, 256) \rightarrow (256, 256, 128)$	SAC*(256, 3, True), Noise, FusedLeakyReLU
	$(256, 256, 128) \rightarrow (256, 256, 128)$	SAC*(256, 3, False), Noise, FusedLeakyReLU
Output Layer	$(256, 256, 128) \rightarrow (256, 256, 3)$	SAC*(256, 3, False)

Table 4: **SMILE Semantically-driven Image Synthesis Generator network architecture.** We leverage on the StyleGAN2 [3] architecture with minor yet significant modifications. We replace the modulated convolutions layers with our Semantically Adaptive Convolutions (SACs) layers. Additionally, for the last three layers (SACs*) we only introduce the semantics in a SPADE fashion. EqualConv2d, Blur, Noise and FusedLeakyReLU are mirror layers from StyleGAN [2]. For the most part of the coding, we heavily borrow from https://github.com/rosinality/stylegan2-pytorch.

Layer	Input → Output Shape	Layer Information
Input Layer	$(256, 256, 3) \rightarrow (256, 256, 32)$	EqualConv2d(dim_out=32, kernel=3, stride=1, padding=1), FusedLeakyReLU
Hidden Layer	$(256, 256, 32) \rightarrow (128, 128, 64)$	Blur, EqualConv2d(64, 3, 2, 0), FusedLeakyReLU
Hidden Layer	$(128, 128, 64) \rightarrow (64, 64, 128)$	Blur, EqualConv2d(128, 3, 2, 0), FusedLeakyReLU
Hidden Layer	$(64, 64, 128) \rightarrow (128, 128, 256)$	EqualConv2d(256, 3, 2, 0, upsample=True), Blur, FusedLeakyReLU
Hidden Layer	$(128, 128, 256) \rightarrow (64, 64, \mathbb{N}_s^{syn})$	Blur, EqualConv2d(512, 3, 2, 0), FusedLeakyReLU
Output Layer	$(64, 64, \mathbb{N}_s^{syn}) \rightarrow (\mathbb{N}_{mask}, \mathbb{N}_s^{syn})$	Mask Average Pooling

Table 5: **SMILE Semantically-driven Image Synthesis Style Encoder network architecture.** Using the RGB and Semantic Mask ($\mathbb{N}_{mask}$: 19) as input, it outputs a per region style with dimentionality $\mathbb{N}_s^{syn}$: 64. EqualConv2D, FusedLeakyReLU and Blur are layers borrowed from StyleGAN [2]. Mask Average Pooling combines the semantic information with the style encoded features.

Layer	Input → Output Shape	Layer Information
Input Layer	$(\mathbb{N}_{mask} \times \mathbb{N}_s^{syn}) \rightarrow (\mathbb{N}_{mask} \times \mathbb{N}_s^{syn})$	PixelNorm
	$(\mathbb{N}_{mask} \times \mathbb{N}_s^{syn}) \rightarrow (\mathbb{N}_{mask} \times \mathbb{N}_s^{syn})$	EqualLinear(dim_out=($\mathbb{N}_{mask} \times \mathbb{N}_s^{syn}$))
	$(\mathbb{N}_{mask} \times \mathbb{N}_s^{syn}) \rightarrow (\mathbb{N}_{mask} \times \mathbb{N}_s^{syn})$	EqualLinear(dim_out=($\mathbb{N}_{mask} \times \mathbb{N}_s^{syn}$))
	$(\mathbb{N}_{mask} \times \mathbb{N}_s^{syn}) \rightarrow (\mathbb{N}_{mask} \times \mathbb{N}_s^{syn})$	EqualLinear(dim_out=($\mathbb{N}_{mask} \times \mathbb{N}_s^{syn}$))
	$(\mathbb{N}_{mask} \times \mathbb{N}_s^{syn}) \rightarrow (\mathbb{N}_{mask} \times \mathbb{N}_s^{syn})$	EqualLinear(dim_out=($\mathbb{N}_{mask} \times \mathbb{N}_s^{syn}$))
	$(\mathbb{N}_{mask} \times \mathbb{N}_s^{syn}) \rightarrow (\mathbb{N}_{mask} \times \mathbb{N}_s^{syn})$	EqualLinear(dim_out=($\mathbb{N}_{mask} \times \mathbb{N}_s^{syn}$))
	$(\mathbb{N}_{mask} \times \mathbb{N}_s^{syn}) \rightarrow (\mathbb{N}_{mask} \times \mathbb{N}_s^{syn})$	EqualLinear(dim_out=($\mathbb{N}_{mask} \times \mathbb{N}_s^{syn}$))
	$(\mathbb{N}_{mask} \times \mathbb{N}_s^{syn}) \rightarrow (\mathbb{N}_{mask} \times \mathbb{N}_s^{syn})$	EqualLinear(dim_out=($\mathbb{N}_{mask} \times \mathbb{N}_s^{syn}$))
	$(\mathbb{N}_{mask} \times \mathbb{N}_s^{syn}) \rightarrow (\mathbb{N}_{mask} \times \mathbb{N}_s^{syn})$	EqualLinear(dim_out=($\mathbb{N}_{mask} \times \mathbb{N}_s^{syn}$))

Table 6: **SMILE Semantically-driven Image Synthesis Mapping network architecture.** The input is sampled from a gaussian distribution, and the output is also known as the style distribution $\mathbb{W}$. PixelNorm and EqualLinear are pixel normalization and linearly normalized layers introduced in StyleGAN [2], respectively.

Attribute Manipulation	CelebA-HQ [1] — Reference Synthesis							
	Pose↓			Attributes↑		Reconstruction↑	Perceptual	
	Pitch	Roll	Yaw	AP	F1	mIoU	FID↓	LPIPS↑
Male	14.812 ± 22.972	1.735 ± 6.619	12.462 ± 24.730	0.916 ± 0.127	0.914 ± 0.143	0.990 ± 0.005	33.197 ± 0.000	0.275 ± 0.036
Female	12.600 ± 32.221	2.250 ± 8.427	12.983 ± 24.233	0.942 ± 0.103	0.939 ± 0.095	0.990 ± 0.004	16.408 ± 0.000	0.281 ± 0.030
Removing Eyeglasses	15.426 ± 30.238	2.036 ± 3.528	9.981 ± 23.557	0.908 ± 0.173	0.892 ± 0.179	0.987 ± 0.004	69.677 ± 0.000	0.059 ± 0.021
Adding Eyeglasses	17.774 ± 25.477	2.388 ± 7.971	9.269 ± 21.661	0.951 ± 0.074	0.922 ± 0.095	0.990 ± 0.004	37.946 ± 0.000	0.102 ± 0.025
Removing Hair	17.561 ± 36.659	2.167 ± 6.065	9.580 ± 22.614	0.927 ± 0.091	0.905 ± 0.090	0.989 ± 0.004	68.845 ± 0.000	0.095 ± 0.052
Adding Hair	12.212 ± 28.076	1.704 ± 2.898	9.933 ± 15.999	0.994 ± 0.009	0.990 ± 0.014	0.990 ± 0.003	110.764 ± 0.000	0.071 ± 0.018
Removing Bangs	8.608 ± 14.954	2.605 ± 9.193	10.471 ± 27.656	0.894 ± 0.173	0.892 ± 0.137	0.988 ± 0.005	33.784 ± 0.000	0.164 ± 0.047
Adding Bangs	10.000 ± 23.522	1.680 ± 5.321	7.913 ± 17.688	0.958 ± 0.065	0.941 ± 0.072	0.990 ± 0.004	25.799 ± 0.000	0.158 ± 0.045
Removing Earrings	6.767 ± 13.369	0.959 ± 1.799	6.655 ± 15.190	0.955 ± 0.073	0.940 ± 0.076	0.989 ± 0.004	52.971 ± 0.000	0.025 ± 0.011
Adding Earrings	12.059 ± 27.609	1.674 ± 5.348	7.656 ± 19.257	0.983 ± 0.028	0.965 ± 0.036	0.990 ± 0.004	35.760 ± 0.000	0.036 ± 0.018
Removing Hat	17.459 ± 30.457	2.531 ± 3.671	8.516 ± 15.985	0.962 ± 0.086	0.952 ± 0.107	0.983 ± 0.008	67.065 ± 0.000	0.087 ± 0.030
Adding Hat	13.427 ± 27.384	1.646 ± 6.487	7.848 ± 17.526	0.910 ± 0.147	0.881 ± 0.166	0.990 ± 0.004	50.865 ± 0.000	0.194 ± 0.042

Table 7: **Additional quantitative results for semantic manipulation using exemplar images.**

Attribute Manipulation	CelebA-HQ [1] — Latent Synthesis							
	Pose↓			Attributes↑		Reconstruction↑	Perceptual	
	Pitch	Roll	Yaw	AP	F1	mIoU	FID↓	LPIPS↑
Male	13.590 ± 22.054	2.072 ± 9.391	13.673 ± 30.232	0.958 ± 0.094	0.942 ± 0.130	0.990 ± 0.005	40.799 ± 0.000	0.418 ± 0.026
Female	11.955 ± 29.208	2.350 ± 9.625	10.281 ± 31.014	0.946 ± 0.116	0.947 ± 0.105	0.990 ± 0.004	22.844 ± 0.000	0.424 ± 0.030
Removing Eyeglasses	22.192 ± 45.327	3.591 ± 8.847	11.065 ± 26.139	0.913 ± 0.167	0.924 ± 0.140	0.987 ± 0.004	45.112 ± 0.000	0.393 ± 0.031
Adding Eyeglasses	21.041 ± 29.497	3.287 ± 9.143	11.461 ± 26.072	0.969 ± 0.055	0.939 ± 0.078	0.990 ± 0.004	42.884 ± 0.000	0.410 ± 0.030
Removing Hair	19.733 ± 33.123	3.052 ± 8.357	11.608 ± 28.834	0.959 ± 0.043	0.929 ± 0.057	0.989 ± 0.004	74.794 ± 0.000	0.388 ± 0.032
Adding Hair	17.214 ± 29.187	2.467 ± 5.099	13.014 ± 46.367	1.000 ± 0.000	1.000 ± 0.000	0.989 ± 0.003	72.738 ± 0.000	0.392 ± 0.026
Removing Bangs	11.708 ± 27.843	2.995 ± 11.400	12.537 ± 33.663	0.985 ± 0.029	0.965 ± 0.054	0.988 ± 0.005	29.660 ± 0.000	0.381 ± 0.033
Adding Bangs	10.996 ± 21.612	2.121 ± 5.462	9.060 ± 22.742	0.981 ± 0.030	0.960 ± 0.046	0.990 ± 0.004	27.725 ± 0.000	0.398 ± 0.031
Removing Earrings	8.376 ± 14.259	1.336 ± 2.549	8.695 ± 28.302	0.989 ± 0.026	0.976 ± 0.050	0.989 ± 0.004	39.078 ± 0.000	0.371 ± 0.033
Adding Earrings	13.158 ± 25.004	2.221 ± 6.837	9.153 ± 23.328	0.997 ± 0.003	0.987 ± 0.010	0.990 ± 0.004	32.287 ± 0.000	0.376 ± 0.032
Removing Hat	15.321 ± 22.200	3.623 ± 5.605	15.344 ± 29.127	0.904 ± 0.188	0.891 ± 0.182	0.983 ± 0.008	49.037 ± 0.000	0.394 ± 0.028
Adding Hat	15.697 ± 25.855	1.949 ± 5.856	9.540 ± 21.746	0.926 ± 0.125	0.896 ± 0.146	0.990 ± 0.004	40.860 ± 0.000	0.442 ± 0.031

Table 8: **Additional quantitative results for semantic manipulation using latent synthesis.**

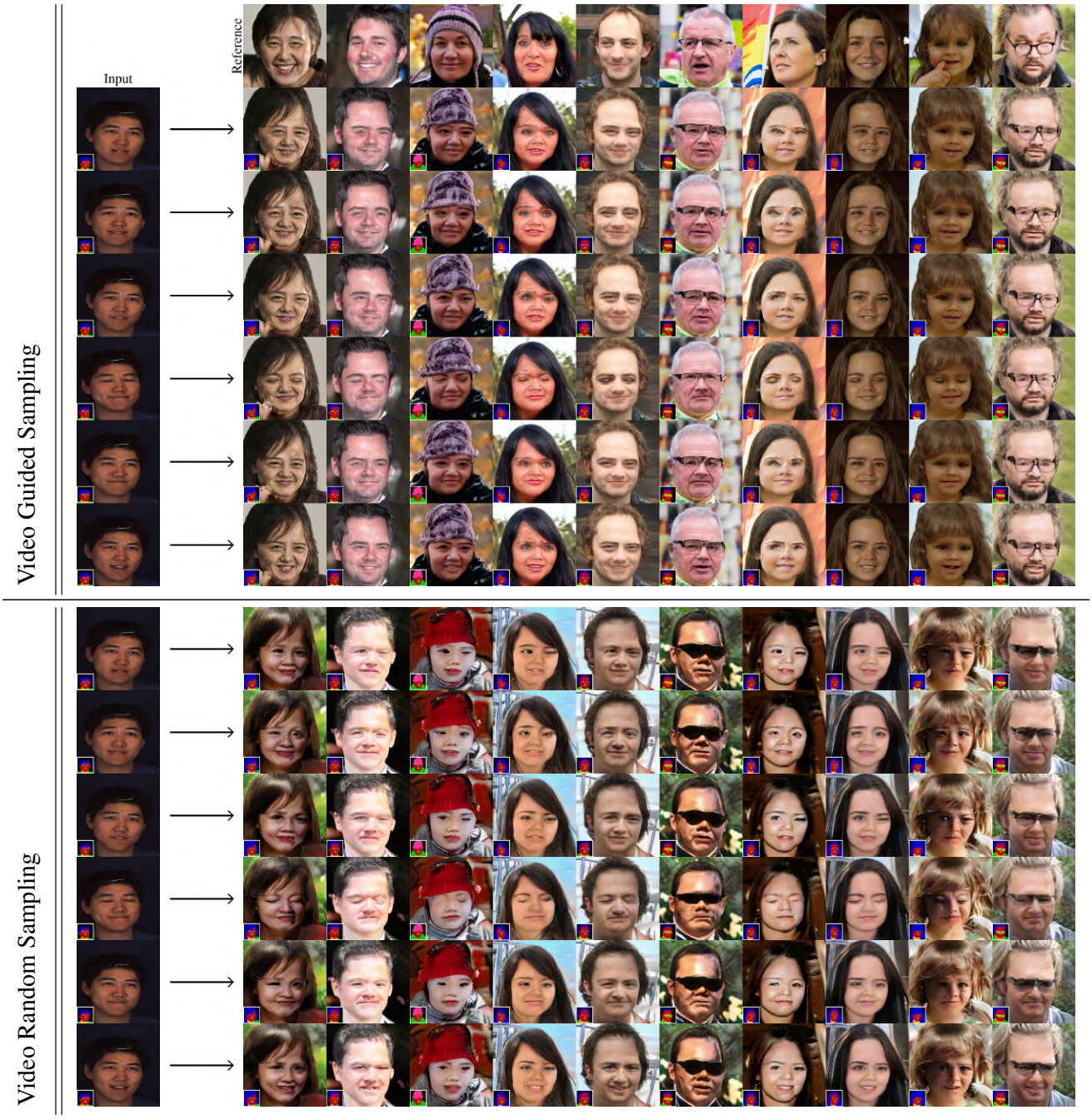
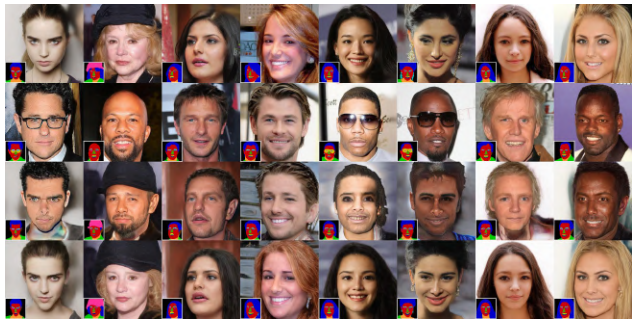
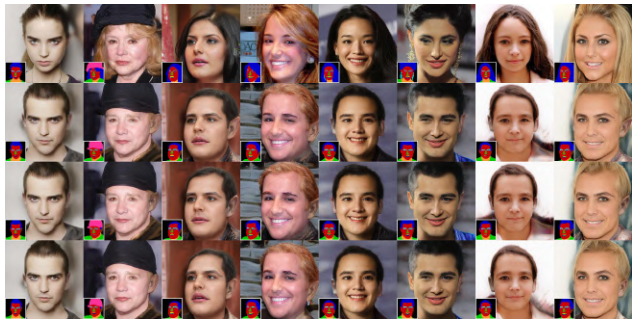


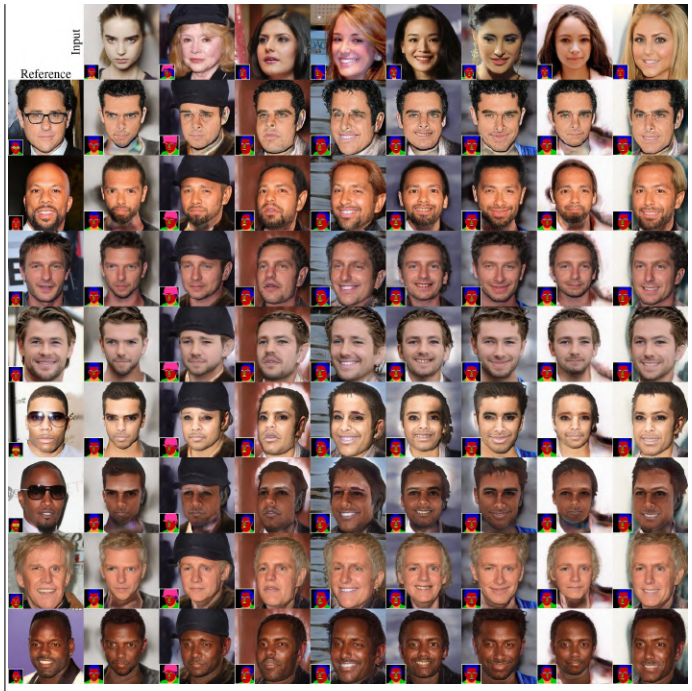
Figure 1: Qualitative visualizations for SMILE. These images highlights the full transformation framework. First, we manipulate the semantic input with respect to the style reference. Second, for both reference and random sampling we keep the same style matrix and the video generation is driven by the semantic feed. Note that upper and lower figures share the same semantics (left bottom corner of each image). Remarkably, we do not train our system using video information.



(a) Reconstruction

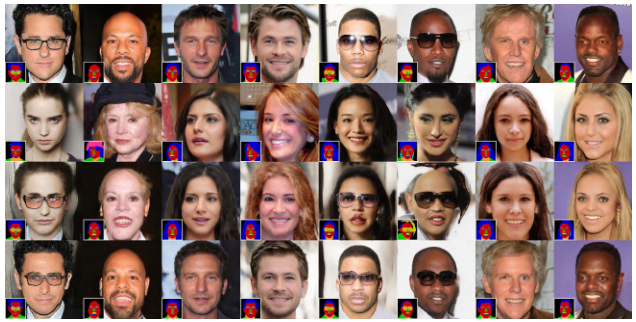


(b) Latent Synthesis



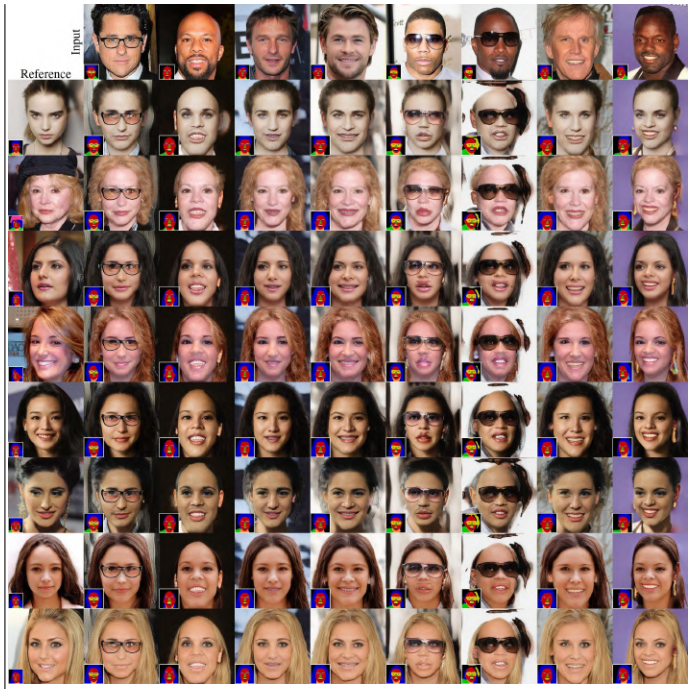
(c) Reference Synthesis

Figure 2: **Qualitative Results for Male manipulation only**. (a) Reconstruction results. First and second row are input and reference images, respectively. Third and fourth rows are forward and reconstruction outputs, respectively. (b) Latent synthesis generation for both semantic and rgb outputs. First row represents input images. (c) reference image synthesis for both semantic and rgb space.



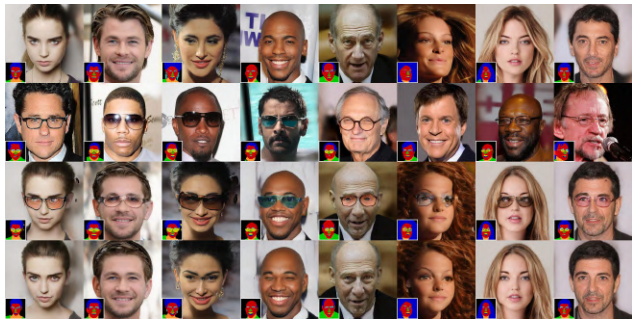
(a) Reconstruction

(b) Latent Synthesis

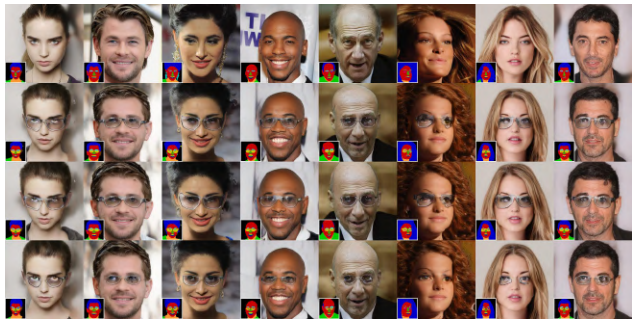


(c) Reference Synthesis

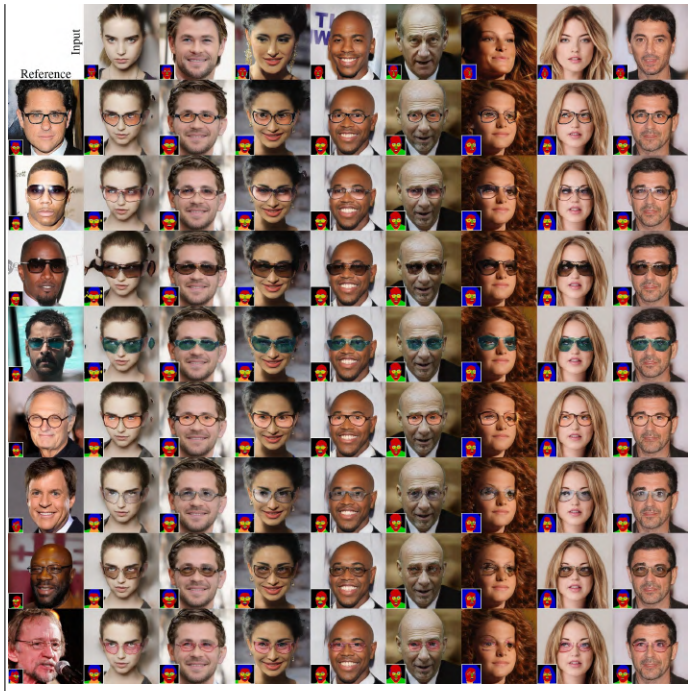
Figure 3: **Qualitative Results for Female manipulation only**. (a) Reconstruction results. First and second row are input and reference images, respectively. Third and fourth rows are forward and reconstruction outputs, respectively. (b) Latent synthesis generation for both semantic and rgb outputs. First row represents input images. (c) reference image synthesis for both semantic and rgb space.



(a) Reconstruction

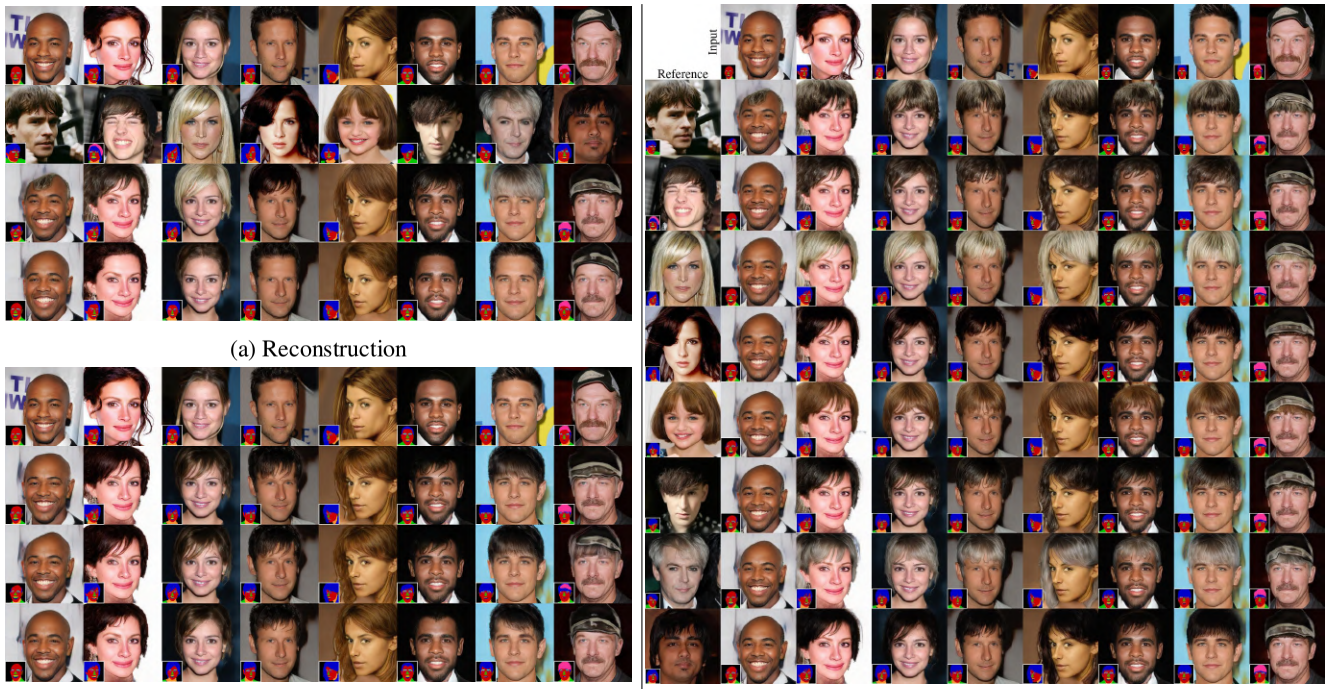


(b) Latent Synthesis



(c) Reference Synthesis

Figure 4: **Qualitative Results for Eyeglasses manipulation only**. (a) Reconstruction results. First and second row are input and reference images, respectively. Third and fourth rows are forward and reconstruction outputs, respectively. (b) Latent synthesis generation for both semantic and rgb outputs. First row represents input images. (c) reference image synthesis for both semantic and rgb space.

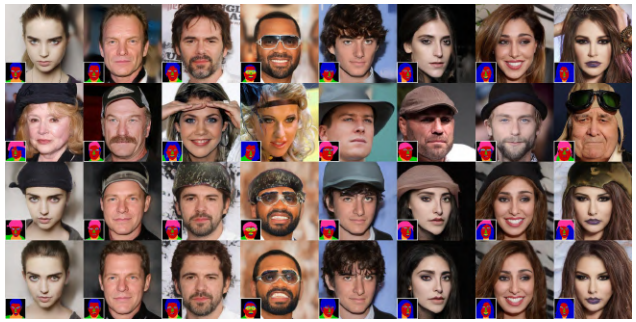


(a) Reconstruction

(b) Latent Synthesis

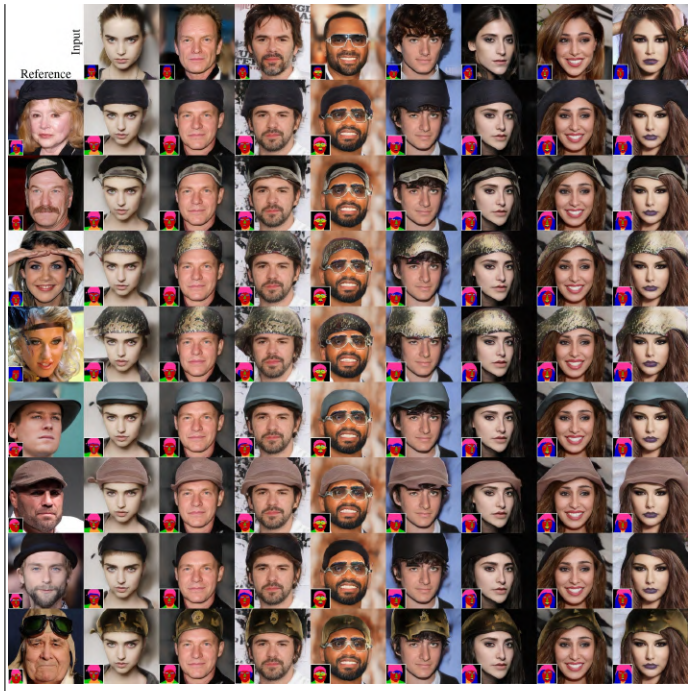
(c) Reference Synthesis

Figure 5: **Qualitative Results for Bangs manipulation only**. (a) Reconstruction results. First and second row are input and reference images, respectively. Third and fourth rows are forward and reconstruction outputs, respectively. (b) Latent synthesis generation for both semantic and rgb outputs. First row represents input images. (c) reference image synthesis for both semantic and rgb space.



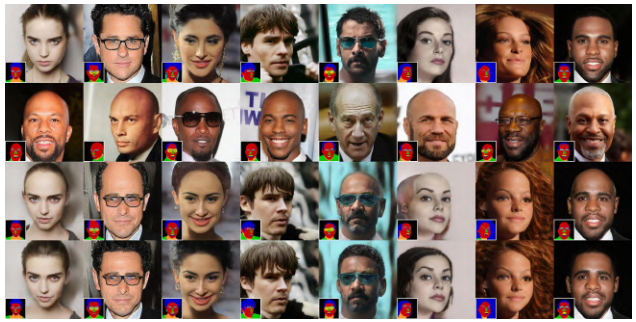
(a) Reconstruction

(b) Latent Synthesis



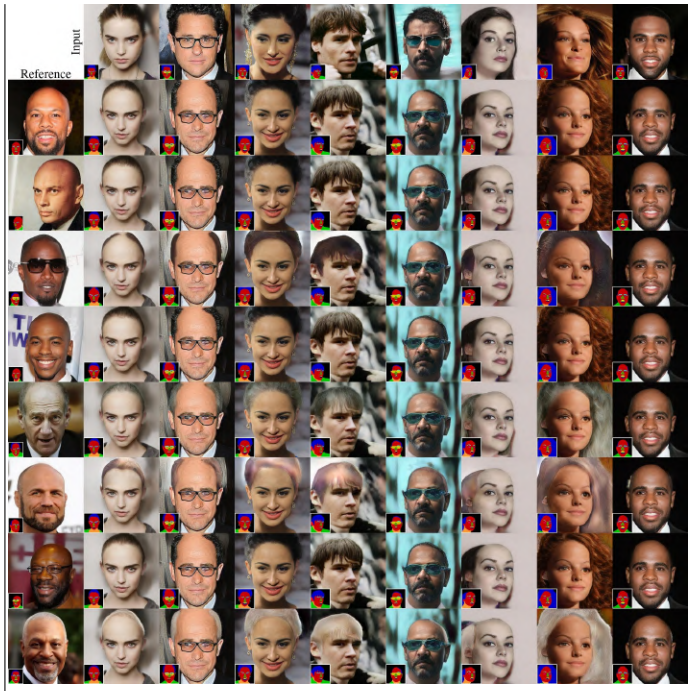
(c) Reference Synthesis

Figure 6: **Qualitative Results for Hat manipulation only**. (a) Reconstruction results. First and second row are input and reference images, respectively. Third and fourth rows are forward and reconstruction outputs, respectively. (b) Latent synthesis generation for both semantic and rgb outputs. First row represents input images. (c) reference image synthesis for both semantic and rgb space.



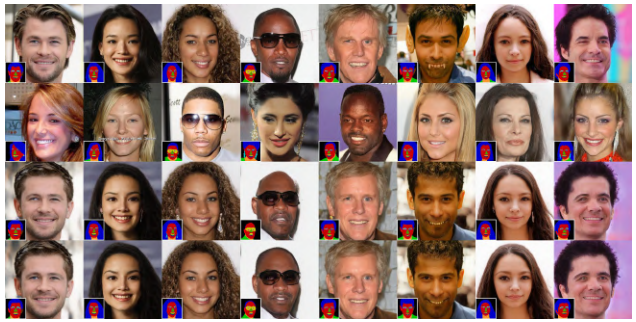
(a) Reconstruction

(b) Latent Synthesis

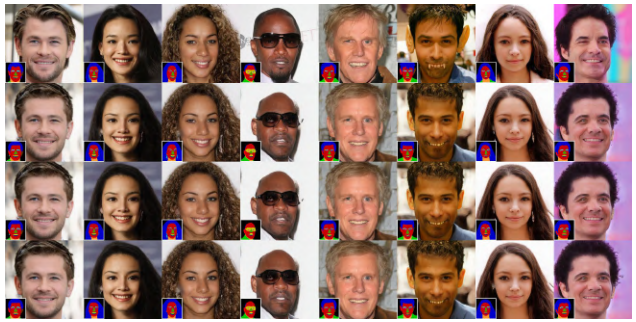


(c) Reference Synthesis

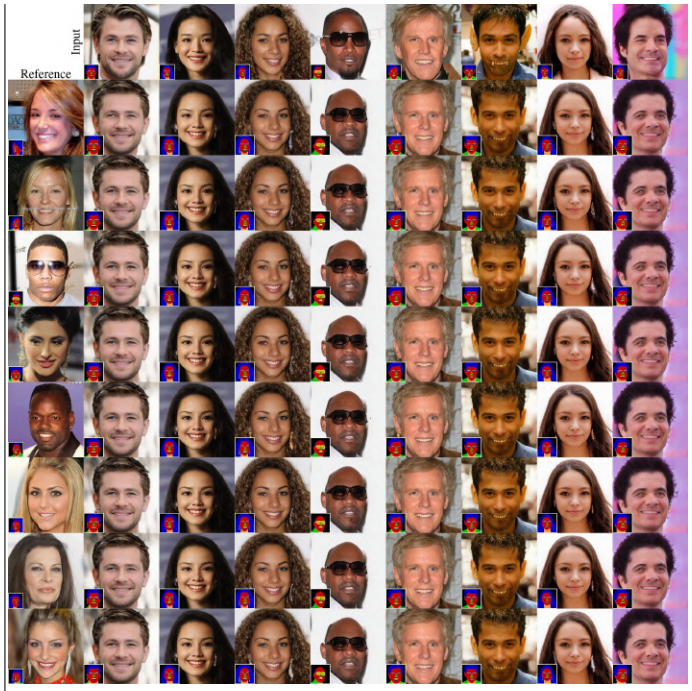
Figure 7: **Qualitative Results for Hair manipulation only**. (a) Reconstruction results. First and second row are input and reference images, respectively. Third and fourth rows are forward and reconstruction outputs, respectively. (b) Latent synthesis generation for both semantic and rgb outputs. First row represents input images. (c) reference image synthesis for both semantic and rgb space.



(a) Reconstruction



(b) Latent Synthesis



(c) Reference Synthesis

Figure 8: **Qualitative Results for Earrings manipulation only**. (a) Reconstruction results. First and second row are input and reference images, respectively. Third and fourth rows are forward and reconstruction outputs, respectively. (b) Latent synthesis generation for both semantic and rgb outputs. First row represents input images. (c) reference image synthesis for both semantic and rgb space.

FFHQ [2]		
Mask Size	FID↓	Diversity↑
4	13.40	0.42 ± 0.04
8	12.10	0.41 ± 0.04
16	12.56	0.42 ± 0.04
32	15.12	0.42 ± 0.04

Table 9: **Quantitative assessment for the size of the mask during image synthesis.**

B.1. Semantic Manipulation Disentanglement

Furthermore, to study the disentanglement during the transformation, we compute Precision vs Recall curves for each attribute manipulation (see Figure 9 and 10).

C. Additional Results for Image Synthesis

C.1. Semantic mask input size

We study the influence of the mask size as starting point of the generator synthesis (see Table 9).

D. Face Reenactment

Face Reenactment results are possible with a slight modification of our system. In this case, there are no domains, so we entirely rely on the transformation of certain regions of the face and keeping unaltered those regions do not related to the puppeteering. In Figure 11 we show qualitative results.

References

[1] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 6

[2] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019. 4, 5, 15

[3] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8110–8119, 2020. 4

[4] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5549–5558, 2020. 2

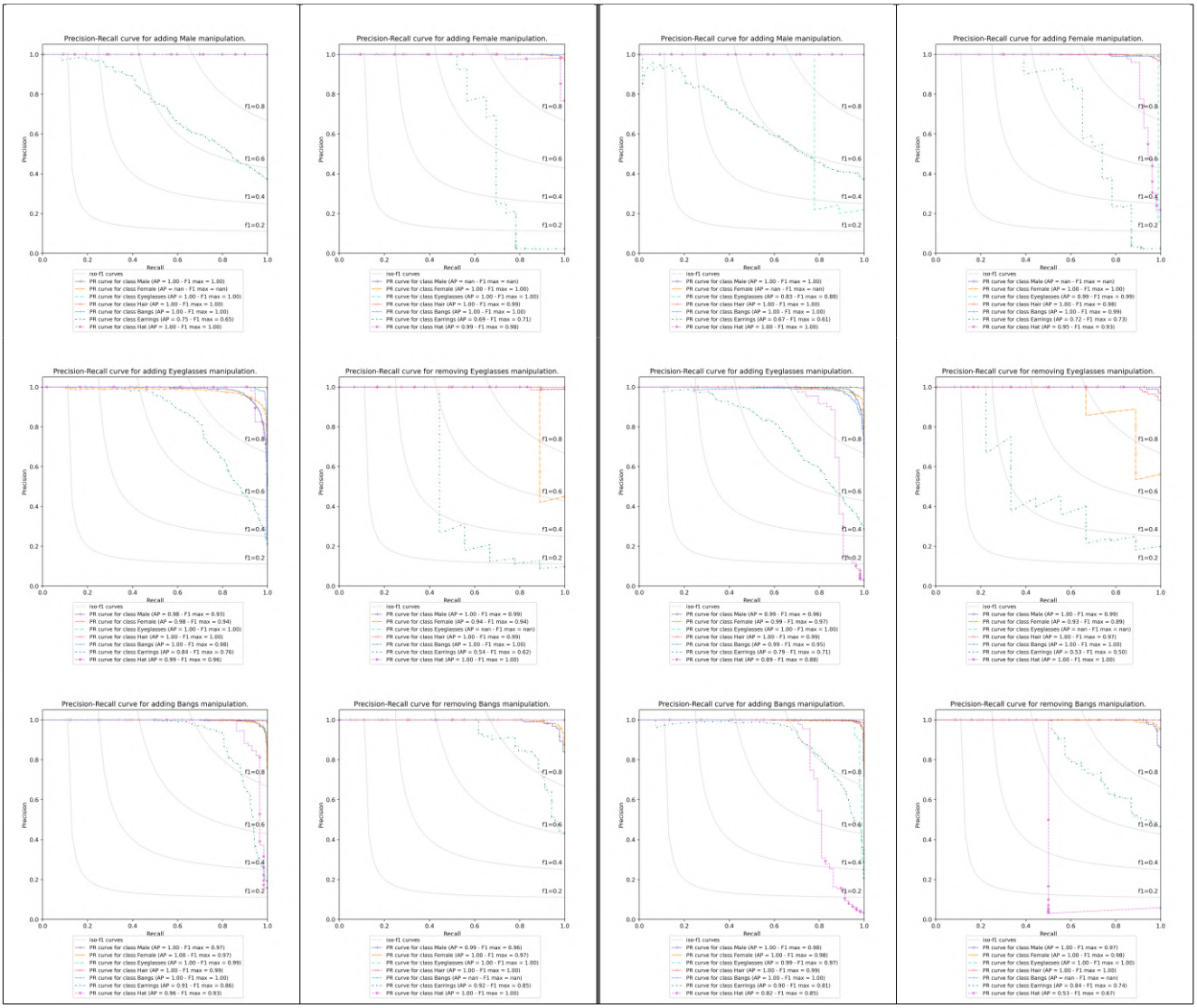


Figure 9: **Precision vs Recall curves for each semantic manipulation for latent styles (left) and reference styles (right).** We study how the prediction of the remaining attributes is affected independently. Please zoom in for better details. Note that NaN in the legend of the images means the there were no images in the test set for the particular manipulation, *e.g*. Male manipulation starts from the Female Test set and transform them to Male, so no Male in the entire set, and hence NaN in the score.

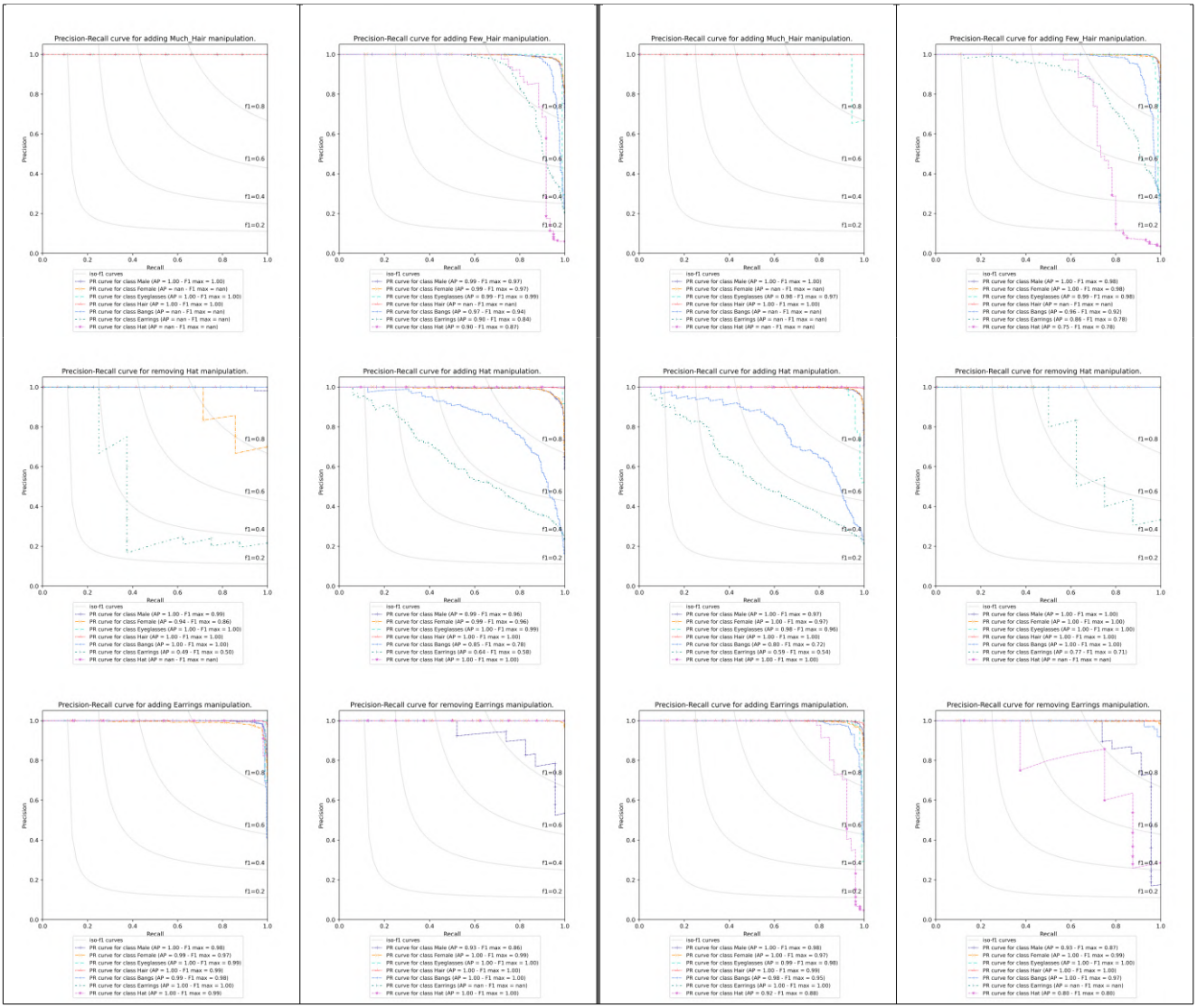


Figure 10: **Precision vs Recall curves for each semantic manipulation for latent styles (left) and reference styles (right).** We study how the prediction of the remaining attributes is affected independently. Please zoom in for better details. Note that NaN in the legend of the images means the there were no images in the test set for the particular manipulation, *e.g.* Much Hair (opposite of bald label) manipulation starts from the Bald label in the Test set and transform them to not Bald, and as only Males with no Earrings and no Hats have bald label in the CelebA-HQ dataset, hence NaN score for Female, Earrings and Hat.

Figure 11: **Face Reenactment Qualitative Results**. By only manipulation the semantic space and keeping the RGB information of the input, we can puppeteer the input with respect to a reference image. Zoom in for better details.