TransEditor: Transformer-Based Dual-Space GAN for Highly Controllable Facial Editing

Supplementary Material

Yanbo Xu[1,4*] Yueqin Yin[1*] Liming Jiang[2] Qianyi Wu[5]
Chengyao Zheng[3] Chen Change Loy[2] Bo Dai[2] Wayne Wu[1,3✉]
[1]Shanghai AI Laboratory [2]S-Lab, Nanyang Technological University [3]SenseTime Research
[4]Hong Kong University of Science and Technology [5]Monash University
yxubu@connect.ust.hk yinyueqin0314@gmail.com qianyi.wu@monash.edu
{liming002, ccloy, bo.dai}@ntu.edu.sg zhengchengyao@sensetime.com wuwenyan0503@gmail.com

1. Detailed Calculation of Metrics

Re-scoring Calculation. This designed metric is used to quantitatively evaluate the editing performance. It is desirable that when editing, the edited attribute will change towards the targeting direction as much as possible, while other attributes remain as less impacted as possible. For example, when editing an attribute towards the plus direction, we expect the score to increase. The amount of change could be quantitatively evaluated using trained classifiers. More specifically, by adding the score difference of the edited attribute between each editing step, the accumulated change could be calculated. Denote the accumulated change of edited attribute as C_e, that of the influenced attribute as C_i, then it is optimal when C_e is as large as possible and C_i is as small as possible. Therefore, the ratio C_i/C_e measures the degree of another attribute being influenced when performing the editing. Note if only C_i is measured as evaluation, the value will be the smallest between two identical images, thus failing to describe the editing performance. Moreover, when the value of C_i are identical, a larger C_e represents more change in the desired attribute, which is desirable for the editing task.

In our experiments, for each attribute, we generate 4,000 images and perform editing. Then the corresponding trained attribute classifier [2] is used to re-score the edited images, resulting in 28,000 scores (6 steps and the origin image) for each attribute.

Identity Re-scoring Calculation. To qualitatively evaluate the change of identity during editing, we also utilize the trained Inception v3 [11] model to extract perceptual features from images. The calculation of this metric is similar to the Re-scoring Calculation above. The cosine similarity of the extracted feature between the images at each step will be calculated, and its accumulated value, denoted as C_{id}, measures the amount of change in identity. The meaning and calculation of C_e are identical to the Re-scoring Calculation. A smaller value of C_{id}/C_e means better preservation of identity when editing the attribute.

Learned Perceptual Image Patch Similarity (LPIPS). LPIPS [12] measures the diversity of a latent space. A larger LPIPS score indicates a more diverse space. Since there are two spaces, we perform this calculation similar to DAT [9]. $\mathbf{LPIPS}_z$ is calculated by sampling 40 $\boldsymbol{z}$ codes with a fixed $\boldsymbol{p}$ code. Similarly, $\mathbf{LPIPS}_p$ is calculated by sampling 40 $\boldsymbol{z}$ codes with a fixed $\boldsymbol{p}$ code. For $\mathbf{LPIPS}_{all}$, it is calculated by sampling 40 pairs of $\boldsymbol{z}$ and $\boldsymbol{p}$ codes. All the processes are repeated by 1,000 times.

Frechet inception distance (FID). FID [4] measures the image generation quality by calculating the feature difference between the real images and the generated images. A smaller FID value implies a better generation quality.

2. Datasets

CelebA-HQ. CelebA-HQ [5] contains 30,000 celebrity face images with a resolution of 1024×1024. The images are annotated with 40 attribute labels.

FFHQ. FFHQ [6] contains 70,000 high-quality face images with a resolution of 1024×1024. FFHQ contains more changes in terms of hue, age, and background than CelebA-HQ.

3. Implementation Details

3.1. Dual Latent Space and Mapping Functions

In our experiments, both the dimension of $\mathcal{Z}$-space and $\mathcal{P}$-space are set to be 16×512. Each latent vector $\boldsymbol{z}_i \in 1 \times 512$ and $\boldsymbol{p}_i \in 1 \times 512$. Their corresponding mapping functions, $M_{\boldsymbol{z}_i}$ and $M_{\boldsymbol{p}_i}$ are MLPs that map $\boldsymbol{z}_i$ to $\boldsymbol{z}_i^+ \in 1 \times 512$ and $\boldsymbol{p}_i$ to $\boldsymbol{p}_i^+ \in 1 \times 512$.

3.2. Transformer-Based Interaction

For interaction at each layer, we utilize a Transformer-based multi-head attention module. In our model, we set

the number of Transformer layers to be 8 and the dimension of the latent code input to Transformer to be 512. The number of heads in the multi-head cross attention module is set to be 8 so the dimensionality for each head is $d_k = 512/8 = 64$. Besides, we add positional encoding to both latent codes from $\mathcal{Z}$-space and $\mathcal{P}$-space before the first Transformer layer. The positional encoding matrix is an identity matrix of size 16×16.

3.3. Training Details

Since two latent spaces are used in the proposed TransEditor, the optimization objective can be written as:

$$\begin{aligned} \min_{\mathbf{G}} \max_{\mathbf{D}} V(\mathbf{D}, \mathbf{G}) = E_{x \sim p_{data}(x)}[\log \mathbf{D}(x)] + \\ E_{(\boldsymbol{z},\boldsymbol{p}) \sim p_{(\mathcal{Z}\times\mathcal{P})}(\boldsymbol{z},\boldsymbol{p})}[\log(1 - \mathbf{D}(\mathbf{G}(\boldsymbol{z}, \boldsymbol{p})))]. \end{aligned} \tag{1}$$

As mentioned, we only apply the adversarial loss [3] and Path Length Regularization used in StyleGAN2 [7]. For the adversarial loss, similar to StyleGAN2, it is composed of non-saturating loss, *i.e.*, $f(t) = softplus(t) = log(1 + exp(t))$. For the generator,

$$L_{\mathbf{G}} = \lambda_{adv} f(-\mathbf{D}(\mathbf{G}(\boldsymbol{z}, \boldsymbol{p}))) + \lambda_{path\_regu} L_{path\_regu}, \tag{2}$$

and L_{path_regu} is the path length regularization. For the discriminator,

$$\begin{aligned} L_{\mathbf{D}} = & \lambda_{dis}[f(\mathbf{D}(X_{fake})) + f(-\mathbf{D}(X_{real}))] \\ & + \lambda_{d\_regu} L_{d\_regu}, \end{aligned} \tag{3}$$

where L_{d_regu} is the gradient regularization for the discriminator.

The training of FFHQ [6] and CelebA-HQ [5] are performed on the resolution of 256×256. We set λ_{d_regu} to 10, L_{path_regu} to 2, λ_{adv} and λ_{dis} to 1 in our training. For CelebA-HQ, we utilize 29,000 images as the training set, with 1,000 left for testing. Then we train the model to 370,000 iterations with a batch size of 16 using 8 cards. For FFHQ, we utilize 69,000 and 1,000 images for training and testing, respectively. The model is trained to 800,000 iterations with the batch size of 16 on a single card.

3.4. Dual Space Inversion and Editing

The loss functions used in our Dual Space Inversion are similar to pSp [10]. We apply the same pixel-wise $\mathcal{L}_2$ loss, LPIPS loss, and ID loss as in pSp [10]. Their weight is set to be 1.0, 0.8, and 0.1, respectively. For both datasets, we train the Dual Space Inversion network to 500,000 iterations, with a batch size of 8.

For Dual Space Editing, auxiliary attribute classifies [2] are used. Specifically, we randomly sample 150,000 pairs of $\boldsymbol{z} \in \mathbb{R}^{n\times512}$ codes and $\boldsymbol{p} \in \mathbb{R}^{n\times512}$ codes from a standard normal distribution and map them to $\boldsymbol{z}^+$ and $\boldsymbol{p}^+$ for image generation. Then for each attribute, the corresponding classifier will be used for scoring on the generated images. We then train an SVM classifier with $\boldsymbol{z}^+$ and the score for attribute i as training data and labels, thus finding the normal vector n_z of the partition interface corresponding to attribute i in $\mathcal{Z}^+$-space. Similarly, we obtain the normal vector n_p in $\mathcal{P}^+$-space. Thereafter, we can move λ_z steps along n_z and λ_p steps along n_p to get the new latent codes $(\boldsymbol{z}^+ + \lambda_z * n_z, \boldsymbol{p}^+ + \lambda_p * n_p)$. We can flexibly adjust λ_z and λ_p to control the contribution of each space to the final editing of different attributes. For example, if we set λ_z to 0, $\mathcal{Z}^+$-space is fixed, and only $\mathcal{P}^+$-space is altered. This diagram can be applied to smile editing (Fig. 8a). Similarly, if we set λ_p to 0, $\mathcal{P}^+$-space is fixed, and only $\mathcal{Z}^+$-space is altered. This diagram can be applied to pose editing (Fig. 5, Fig. 6). For gender (Fig. 7a) and age (Fig. 9b) editing, both λ_z and λ_p are adjusted.

4. More Results and Analysis

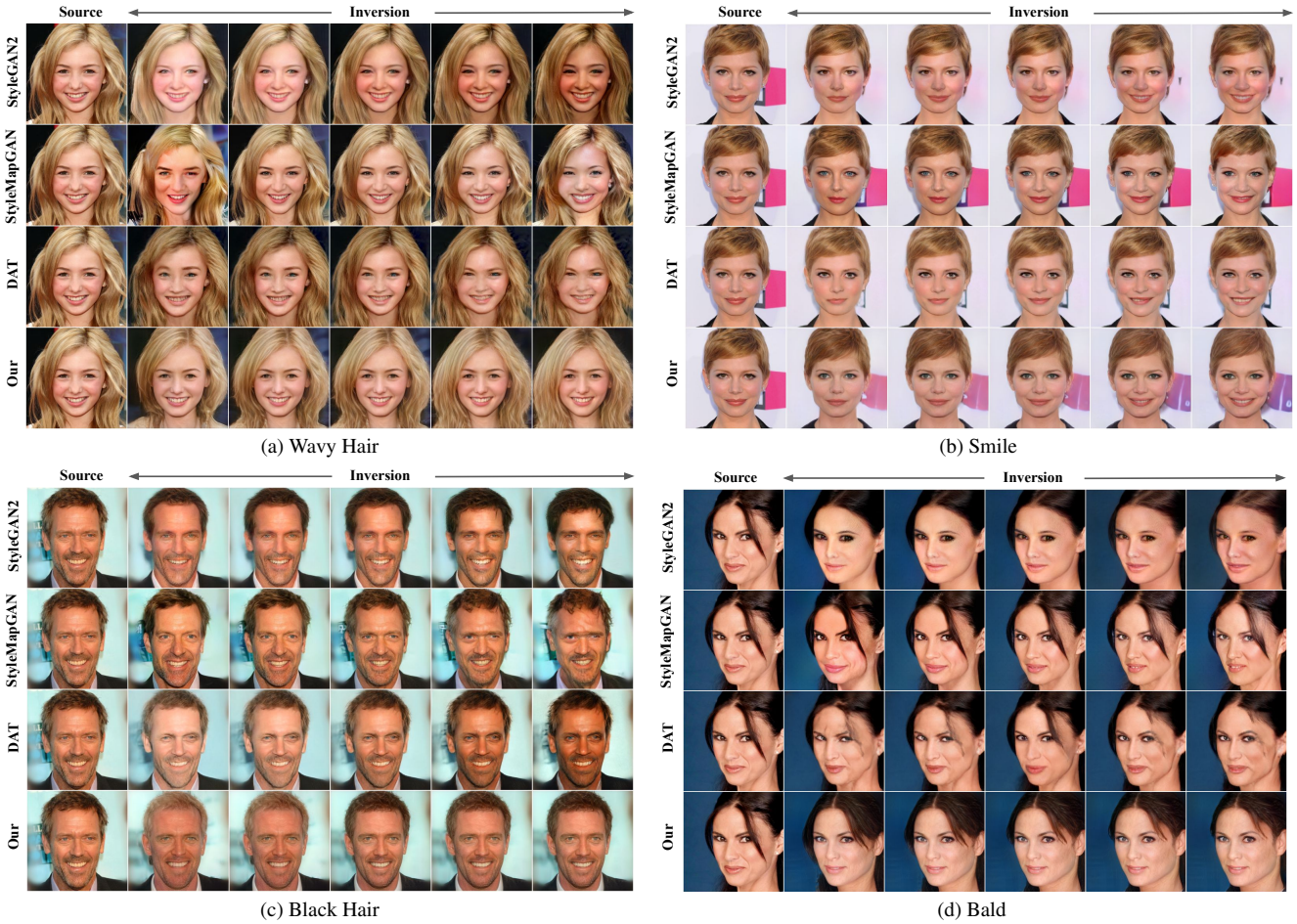
4.1. More Ablation Study

Trials of Training Techniques. The Path Length Regularization in StyleGAN2 [7] is used to smooth the latent space $\mathcal{W}$, which is minimized when changing a fixed step of the latent code will result in a fixed-magnitude change in the image. In our design, the space that corresponds to $\mathcal{W}$ in StyleGAN2 [7] is the output of our cross-space interaction. Since it is not desired that any certain layer of the $\mathcal{W}$ space be dominant, we utilize the same regularization loss on our $\mathcal{W}$ space.

We have also experimented to add the regularization loss on the $\mathcal{P}^+$-space. The result shows that this will discourage the $\mathcal{P}^+$ to influence the final result. When this regularization loss is added on the $\mathcal{P}^+$-space, changing the entire $\boldsymbol{p}$ code will only result in a little change in the generated image, which is undesirable since we need more balanced dual spaces for editing.

Number of Transformer. The number of Transformer layers is related to the degree of interaction. When the number of layers gets larger, the $\boldsymbol{z}^+$ code will be queried by the p^+ code more, resulting in a stronger correlation. We experimented with different number of Transformer, and the result shows that using more layers will result in better head pose consistency when p^+ code is fixed, although the difference is not significant. In most of our settings, we utilize 8 layers of Transformers.

Alternative interaction module (*e.g.*, MLP). The ablation study on Space Interaction via Transformer has shown its cruciality. Our design of using $\mathcal{P}$-space as the query establishes a connection between the two spaces while ensuring their disentanglement. We have also tried other interaction modules, *e.g.*, MLP. Although MLP can create a linkage, the results of which are inferior to the Transformer-based interaction module due to the entanglement caused by MLP. For instance, the Pose-Identity re-scoring (↓) result of the MLP variant is 7.024 compared with our result of **2.326**



(a) Wavy Hair (b) Smile

(c) Black Hair (d) Bald

Figure 1. **Real Image Editing Comparison**. Images on the first column are the real source images. The fourth column shows the reconstruction results, which are semantically interpolated to the left and right sides.

Table 1. **Re-scoring metrics of w/o or w/ Transformer**.

	Pose↓		Gender↓	
Method	w/o Transformer	Ours	w/o Transformer	Ours
Pose	-	-	0.504	**0.231**
Gender	0.782	**0.313**	-	-

Quantitative comparison of editing w/o or w/ Transformer. To further show the importance of the cross-space interaction, a similar re-scoring evaluation is conducted on our dual-space model w/o and w/ Transformer. Tab. 1 shows that adding the Transformer to introduce interaction between latent spaces clearly improves the editing performance.

4.2. More Comparison with State of the Art

Fig. 1 shows more editing results of real images compared with other methods. StyleMapGAN [8] is prone to face distortion when editing attributes. DAT [9] and StyleGAN2 [7] are prone to hue changes. Our model TransEditor achieves the best editing performance. In Fig. 2, we provide an additional comparison with $\boldsymbol{w}^+$ space (Image2StyleGAN [1]) using the same optimization approach for inversion. Our method still achieves the best results.

Table 2. **Identity Re-scoring Calculation.** Compared between StyleGAN2 [7], StyleMapGAN [8], DAT [9], and the proposed TransEditor (Ours).

	ID↓			
Method	StyleGAN2	StyleMapGAN	DAT	Ours
Pose	7.528	25.668	28.693	**2.326**
Gender	1.240	1.209	1.323	**1.135**

Tab. 2 shows the results of Identity Re-scoring on complicated attributes pose and gender, compared with other methods. For DAT [9] and TransEditor, the pose is edited using content space and $\mathcal{P}$-space, respectively, since the structure information is contained in those spaces. Gender is edited using two spaces simultaneously. The result in the first row shows the identity preservation during pose editing. Our method surpasses others by a large margin. This observation is consistent with our quantitative observation on the pose editing results in the main text. Tab. 3 shows the FID metric compared with other methods.

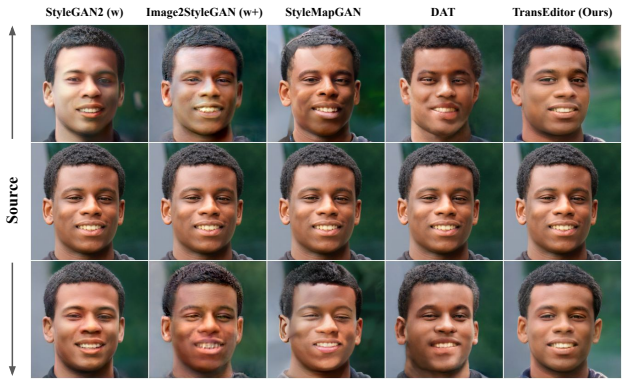


Figure 2. **Pose editing comparison.**

Table 3. **FID Comparison.** All method are trained on FFHQ at the resolution of 256.

Method	**FID** ↓
StyleGAN2 [7]	4.44
StyleMapGAN [8]	15.9
DAT [9]	22.50
Ours	9.32

4.3. More Visualization Results

Similar to the performance on the CelebA-HQ dataset [5], our dual latent spaces also achieve a certain degree of semantic separation on the FFHQ dataset [6], with $\mathcal{P}$-space controlling structural information like pose and $\mathcal{Z}$-space controlling texture information (see Fig. 3, Fig. 4).

The remaining figures show more editing results of TransEditor on different attributes and different datasets. Fig. 5 and Fig. 6 are the pose editing results on the two datasets. Only $\mathcal{P}$-space is used for pose editing. Gender editing results are shown in Fig. 7a and Fig. 7b. As mentioned in the main text, the editing of gender utilizes both spaces. Fig. 8 shows the smile and wavy hair editing on CelebA-HQ [5], they are performed on $\mathcal{Z}$-space and $\mathcal{P}$-space respectively. Fig. 9a shows the results of black hair editing using $\mathcal{Z}$-space on CelebA-HQ [5], and Fig. 9b shows the results of age editing on FFHQ [6]. Since change of age might involve both structure and texture variation, the editing of age is accomplished using both $\mathcal{P}$-space and $\mathcal{Z}$-space simultaneously.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *ICCV*, 2019. 3

[2] Lucy Chai, Jun-Yan Zhu, Eli Shechtman, Phillip Isola, and Richard Zhang. Ensembling with deep generative views. In *CVPR*, 2021. 1, 2

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 27, 2014. 2

[4] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017. 1

[5] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 1, 2, 4

[6] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 1, 2, 4

[7] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 2, 3, 4

[8] Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. Exploiting spatial dimensions of latent in gan for real-time image editing. In *CVPR*, 2021. 3, 4

[9] Gihyun Kwon and Jong Chul Ye. Diagonal attention and style-based gan for content-style disentanglement in image generation and translation. In *ICCV*, 2021. 1, 3, 4

[10] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *CVPR*, 2021. 2

[11] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 1

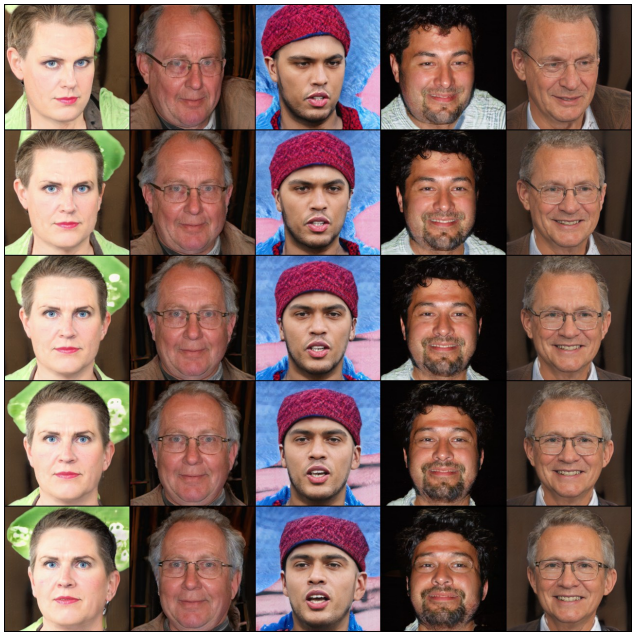
[12] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 1

(a) Fix the $\boldsymbol{p}$ code and sample $\boldsymbol{z}$ code

(b) Fix the $\boldsymbol{z}$ code and sample $\boldsymbol{p}$ code

Figure 3. **Example of disentanglement of dual latent spaces on the FFHQ-256 dataset.** Each column in (a) is generated by a fixed $\boldsymbol{p}$ code and a randomly sampled $\boldsymbol{z}$ code. Note that re-sampling the $\boldsymbol{z}$ code would not influence the head pose. Similarly, for (b), each column shares the same $\boldsymbol{z}$ code. The images generated bare similar lighting, hair color, skin color. This shows the semantic disentanglement of dual latent spaces of TransEditor.

(a) Interpolate $\boldsymbol{z}$ code



(b) Interpolate $\boldsymbol{p}$ code

Figure 4. **Examples from interpolated dual space latent codes on the FFHQ-256 dataset.** In sub-figure (a), each raw has the same $\boldsymbol{z}$ code and interpolated towards the same direction. Each column has the same sampled $\boldsymbol{p}$ code. Notice that the interpolation of the style code gradually changes the hair color, background, and minor facial expression changes without having any effect on the person's head pose. Similarly, for the sub-figure (b), each column shares the same $\boldsymbol{z}$ code and each row shares the same $\boldsymbol{p}$ code. The interpolation of the $\boldsymbol{p}$ code changes the pose along the same direction. Notice that during interpolation, the texture information remains similar.

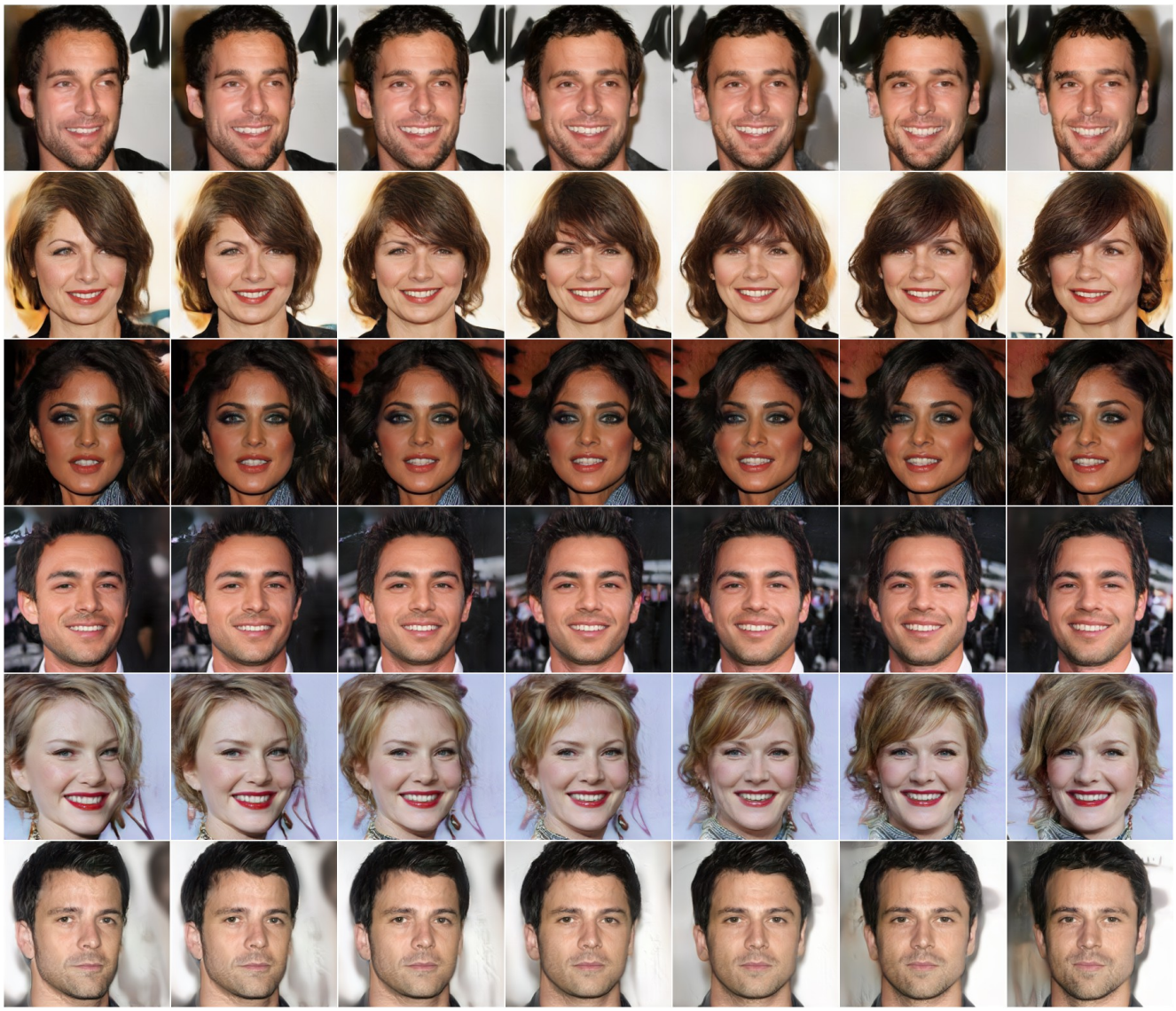


Figure 5. **Pose editing of the sampled images on CelebA-HQ-256 dataset.** Images on the fourth column are the sampled source images, which are semantically interpolated to the left and right sides.

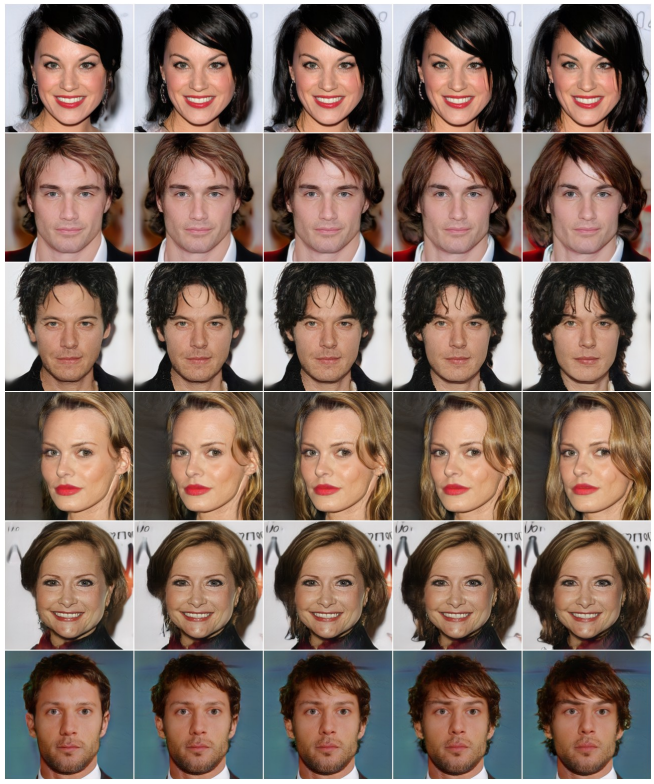
Figure 6. **Pose editing of the sampled images on FFHQ-256 dataset.** Images on the fourth column are the sampled source images, which are semantically interpolated to the left and right sides.

(a) Gender editing on FFHQ-256 dataset.

(b) Gender editing on CelebA-HQ-256 dataset.

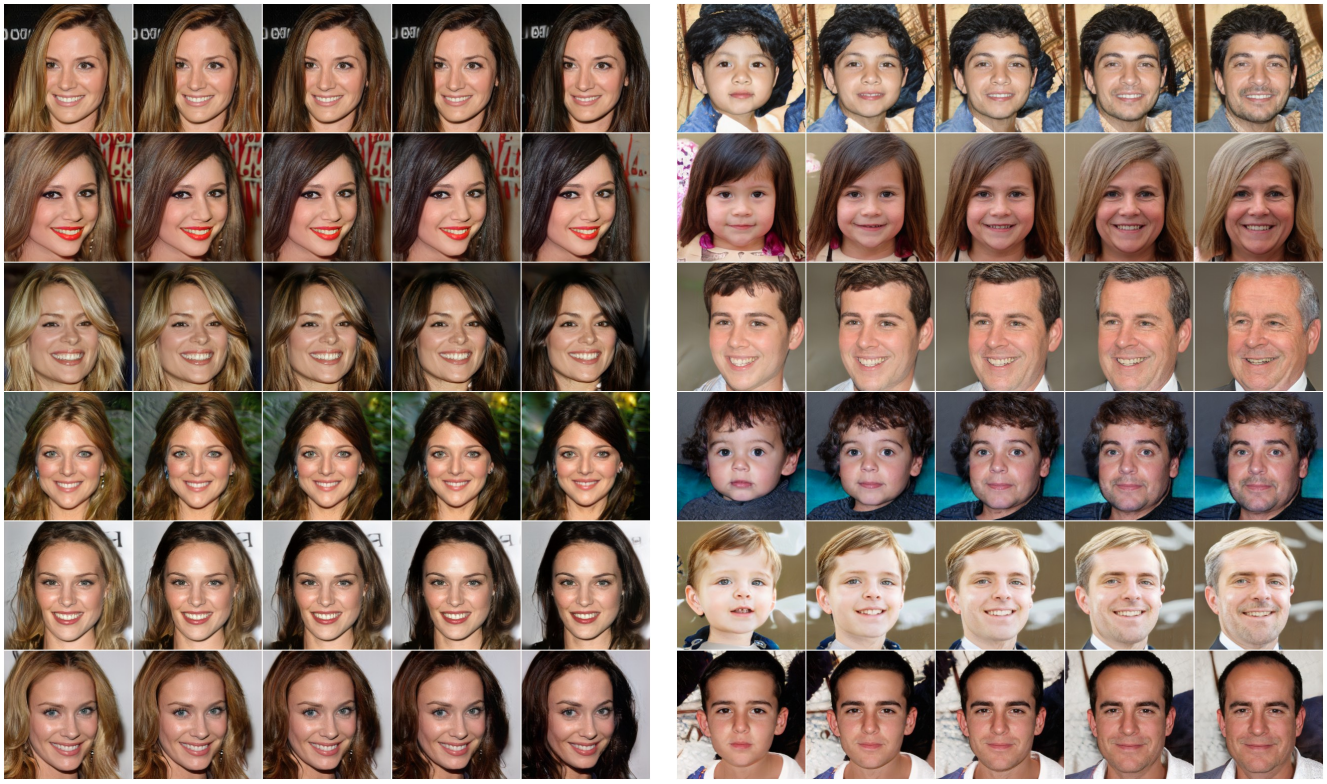
Figure 7. **Gender editing of the sampled images on FFHQ-256 dataset (a) and CelebA-HQ-256 dataset (b).** Images on the third column are the sampled source images, which are semantically interpolated to the left and right sides.

(a) Smile editing on CelebA-HQ-256 dataset.



(b) Wavy Hair editing on CelebA-HQ-256 dataset.

Figure 8. **Sampled Image Editing on CelebA-HQ-256 dataset.** Images on the third column are the sampled source images, which are semantically interpolated to the left and right sides.



(a) Black Hair editing on CelebA-HQ-256 dataset.

(b) Age editing on FFHQ-256 dataset.

Figure 9. **Sampled image editing on CelebA-HQ-256 dataset (a) and FFHQ-256 dataset (b).** Images on the third column are the sampled source images, which are semantically interpolated to the left and right sides.