# StyleGANEX: StyleGAN-Based Manipulation Beyond Cropped Aligned Faces

Shuai Yang Liming Jiang Ziwei Liu Chen Change Loy
S-Lab, Nanyang Technological University
{shuai.yang, liming002, ziwei.liu, ccloy}@ntu.edu.sg

**Supplementary Material**

## Contents

# 1. Implementation Details of StyleGANEX

## 1.1. Dataset and Model

**Dataset.** FFHQ [5] is made available under CC BY-NC-SA 4.0 License by NVIDIA Corporation. FaceForensics++ [7] is released under the FaceForensics Terms of Use at https://kaldir.vc.in.tum.de/faceforensics_tos.pdf. Unsplash (https://unsplash.com/) and Pexels (https://www.pexels.com/) photos and videos are made to be used freely.

**Model**. We build our model based on the PyTorch version of StyleGAN (https://github.com/rosinality/stylegan2-pytorch) and pSp [6] under MIT License. Pix2pixHD [10] is under BSD License. VToonify [12] are under S-Lab License 1.0. TSIT [4] is under CC BY-NC-SA 4.0 License. InterFaceGAN [8], HyperStyle [1] and StyleGAN-NADA [3] are under MIT License. Editing vectors of LowRankGAN [13] are provided at https://github.com/zhujiapeng/LowRankGAN without claiming licenses.

## 1.2. Network Architecture

PSp [6] has multi-scale intermediate layers, where 1-3 layers are for $128 \times 128$ features, the middle 4-7 layers are for $64 \times 64$ features and the subsequent 8-21 layers are for $32 \times 32$ features. For StyleGANEX encoder, we concatenate three features from layers 11, 16 and 21 and add a convolution layer to map the concatenated features to the first-layer input feature $f$.

For skip connections, we use the features from layers 0 (the layer before the intermediate layers), 3, 7, 11, 16, 21, 21 as the skipped features into the StyleGANEX. These seven features are skipped to the StyleGANEX layers corresponding to the resolution 256, 128, 64, 32, 16, 8, 4 of StyleGAN, respectively (corresponding to $\ell$ =13, 11, 9, 7, 5, 3, 1). The skipped feature and the StyleGANEX feature are concatenated and go through an added convolution layer to obtain the fused feature to have the same resolution and channel numbers as the original StyleGANEX feature.

## 1.3. Running Time

**Training encoder** uses one NVIDIA Tesla V100 GPU for 100,000 iterations for all tasks except that video toonification uses 50,000 iterations. The training time is about 2 days for 100,000 iterations and 1 day for 50,000 iterations, respectively.

**Image Inference** uses one NVIDIA Tesla V100 GPU and a batch size of 1. All the following running time includes IO and face detection. Inferencing on 796 testing images which is of averaged $360 \times 398$ size (the corresponding output image is about $1440 \times 1592$), the inversion of each image is about 107.11 s, where the fast feed-forward Step I takes about 0.386 s. For other fast feed-forward tasks such as super-resolution and translation take about 0.259 s-0.545 s depending on the network architectures (*i.e*. how many skip connection layers, whether using $T$).

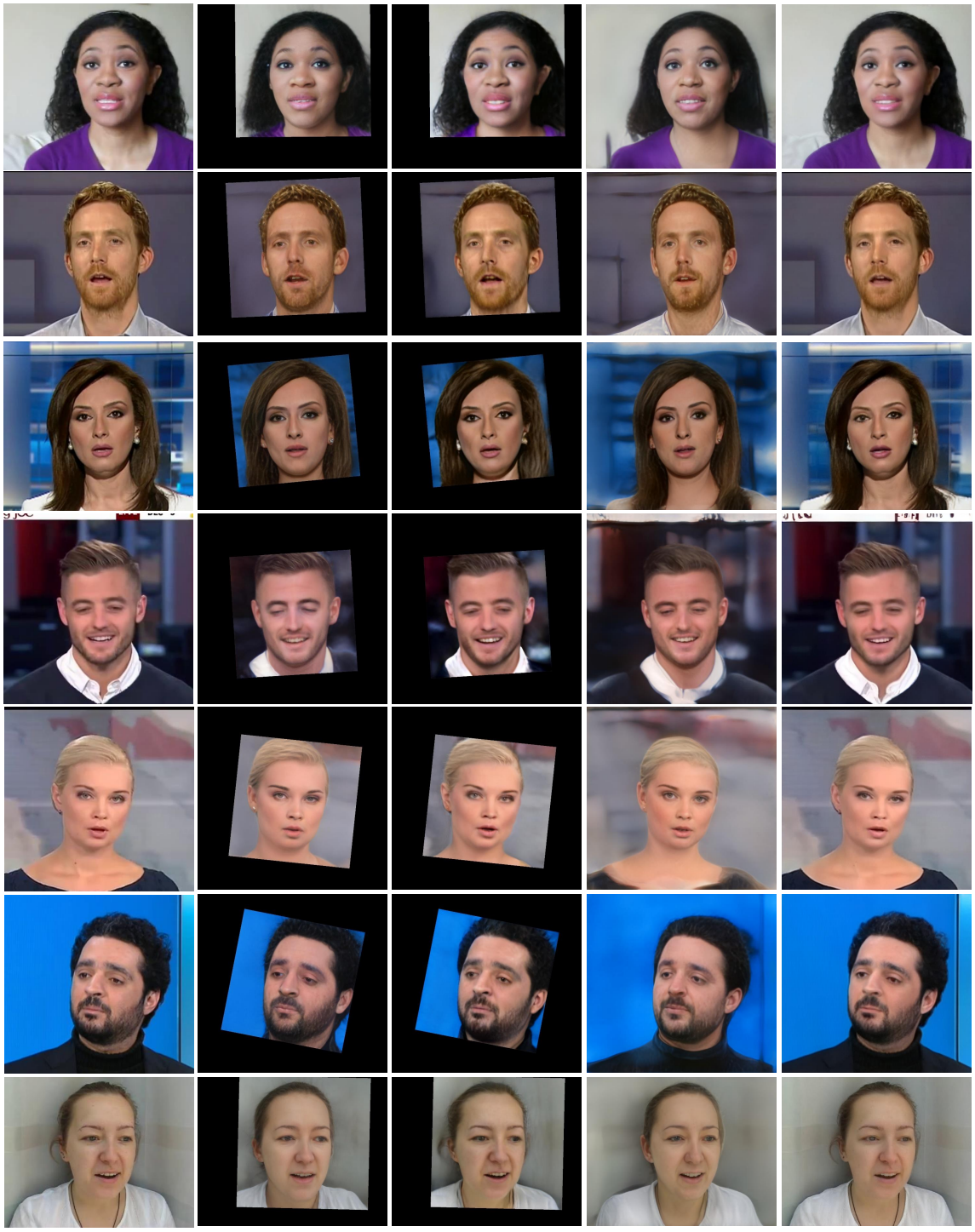
**Video Inference** uses one NVIDIA Tesla V100 GPU and a batch size of 4. All the following running time includes IO and face detection. Inferencing on 28 ten-second video clips, which is of averaged $338 \times 398$ size (the corresponding edited video is about $1352 \times 1592$), the video editing/toonification takes about 45 s per video.

# 2. Supplementary Experimental Results of StyleGANEX

## 2.1. Qualitative Evaluation

In addition to the examples shown in the main paper, we show more visual comparison results in Figs. 1-9.

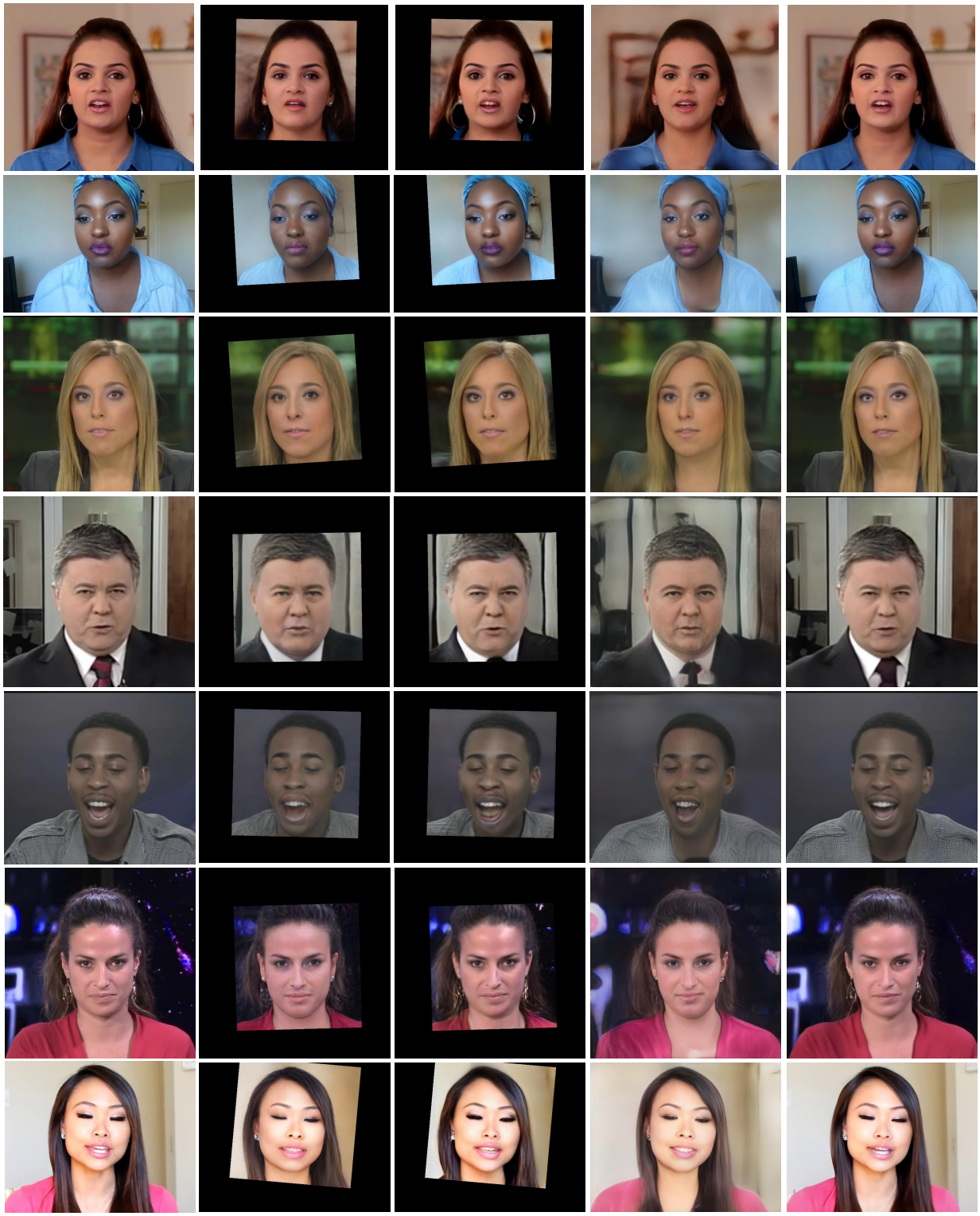
### 2.1.1 Normal FoV face inversion



(a) input (b) pSp (c) HyperStyle (d) our (only Step I) (e) ours

Figure 1. **Comparison on normal FoV face inversion** (Part I).

Figures 1-2 compare with pSp [6] and HyperStyle [1] on normal FoV face inversion. Our encoder surpasses the baselines in handling the complete scenes. With Step-II optimization, our method can further precisely reconstruct the details.

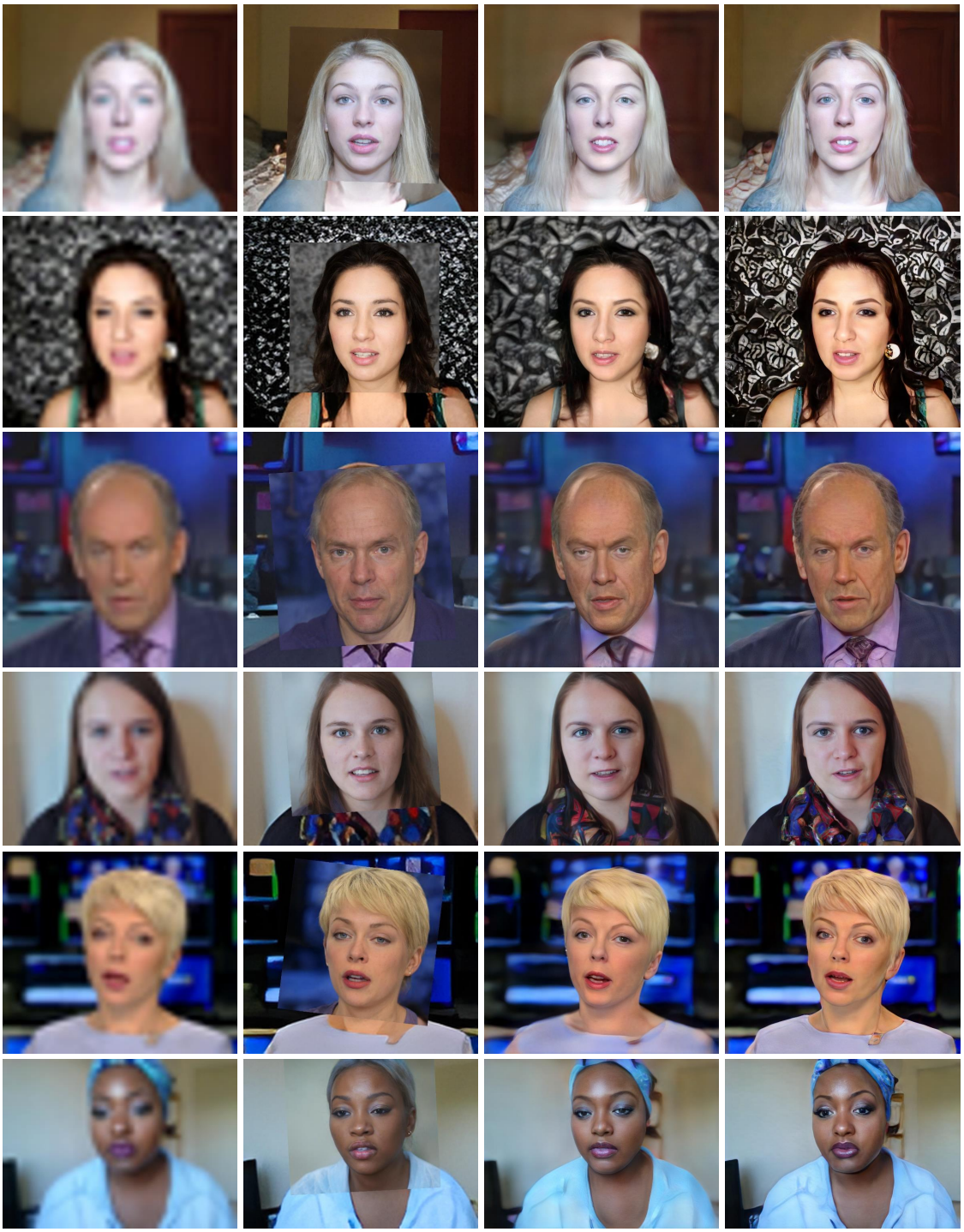


(a) input (b) pSp (c) HyperStyle (d) our (only Step I) (e) ours

Figure 2. **Comparison on normal FoV face inversion** (Part II).

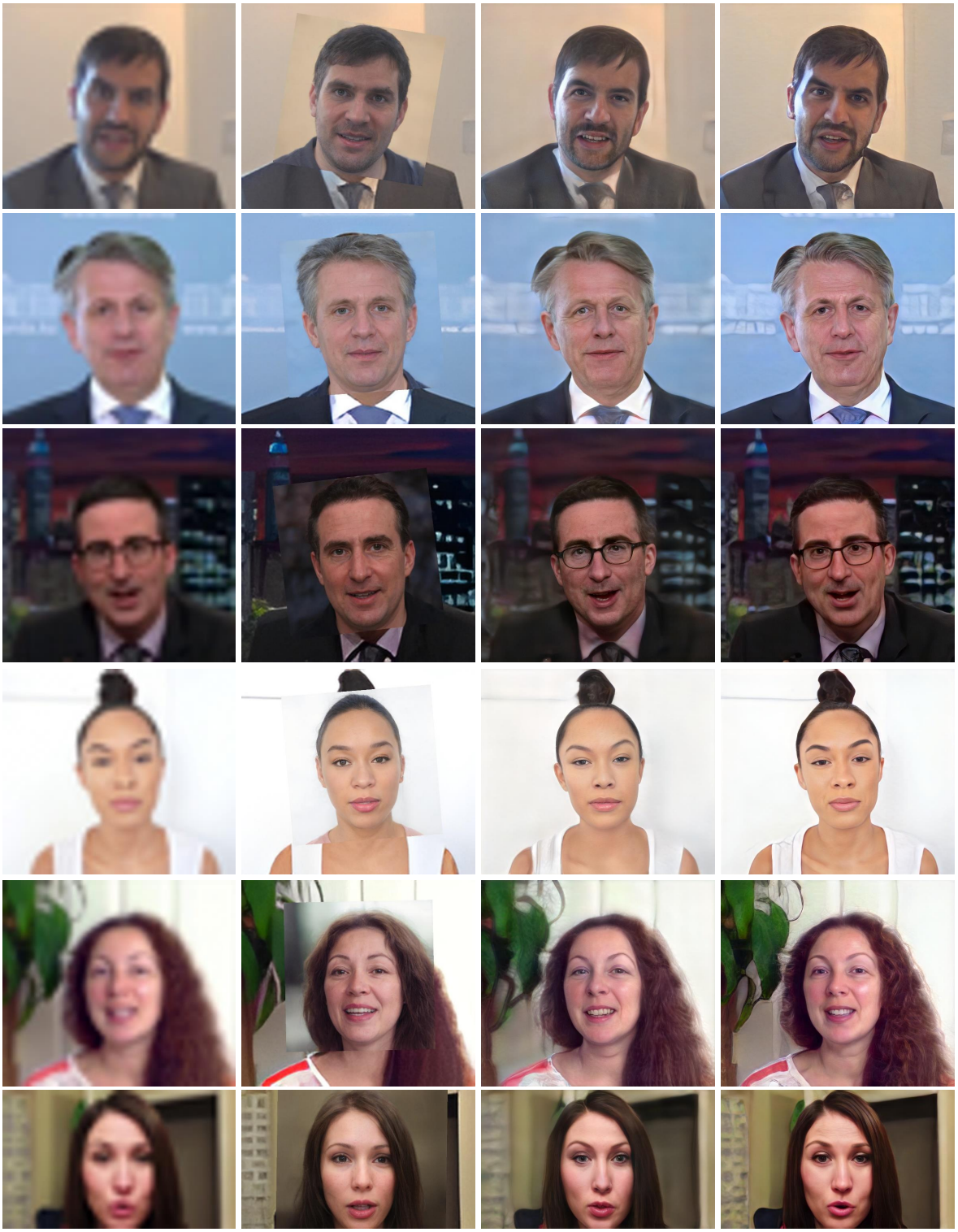
### 2.1.2 Normal FoV face super-resolution

We show $32\times$ super-resolution results in Figs. 3-4(d), where both the face and non-face regions are reasonably restored. We follow pSp to train a single mode on multiple rescaling factors ($4 \sim 64$) with $\ell = 3$ to make a fair comparison. In pSp's results, non-face regions are super-resolved by Real-ESRGAN [11]. As in Figs. 3-4(b)(c), our method surpasses pSp in detail restoration (*e.g.*, glasses) and uniform super-resolution without discontinuity between face and non-face regions.



(a) input (b) pSp (c) ours (d) ours-32

Figure 3. **Comparison on super-resolution** (Part I).



(a) input (b) pSp (c) ours (d) ours-32

Figure 4. **Comparison on super-resolution** (Part II).

### 2.1.3 Sketch/mask-to-face translation

We compare our method with image-to-image translation models pix2pixHD [10] and TSIT [4], and StyleGAN-based pSp in Figs. 5-6. Pix2pixHD's results have many artifacts and monotonous colors. TSIT requires the inputs' side lengths to be divisible by 128. We find padding the input leads to failed translation. Therefore, we show its results on centrally cropped inputs, which are blurry. PSp generates realistic results, which are however less similar to the input sketch/mask. By comparison, our method can translate whole images and achieve realism and structural consistency to the inputs. Furthermore, our method supports multi-modal face generation by sampling style latent codes in the deep 11 layers.


Figure 5. **Comparison on sketch-to-face translation**.

Compared to our method, pix2pixHD [10] pays more attention to keep consistency with the input sketches or masks. For high-quality inputs, pix2pixHD sometimes has overall good translation results (*e.g.*, the third example in Fig. 5). This is why the superiority of our method is not evident in the user study (Table 1) of our main paper. However, for low-quality inputs, our method will be more robust than pix2pixHD. In addition, our method can use adaptive $\ell$ to meet the practical requirement (*i.e.*, more consistency or more robust) as analyzed in Fig. 16 of our main paper.


Figure 6. **Comparison on mask-to-face translation**.

### 2.1.4 Video face attribute editing

Figure 7 compares with pSp and HyperStyle on video face attribute editing. It can be seen that the results of the baselines have discontinuity near the seams. In addition, the latent code alone cannot ensure temporal consistency in videos. The hair details randomly vary without consistency. By comparison, our method uses the first-layer feature and skipped mid-layer features to provide spatial information, which achieves more coherent results (please refer to supplementary videos).


Figure 7. **Comparison on video face attribute editing**.

### 2.1.5 Video face toonification

Figures 8-9 compare with VToonify-T [12] where our method preserves more details of the non-face region and generates sharper faces and hairs. The reason is that VToonify-T uses a fixed latent code extractor while our method trains a joint latent code and feature extractor, thus our method is more powerful for reconstructing the details. Moreover, our method retains StyleGAN's shallow layers, which helps provide key facial features to make the stylized face more vivid.

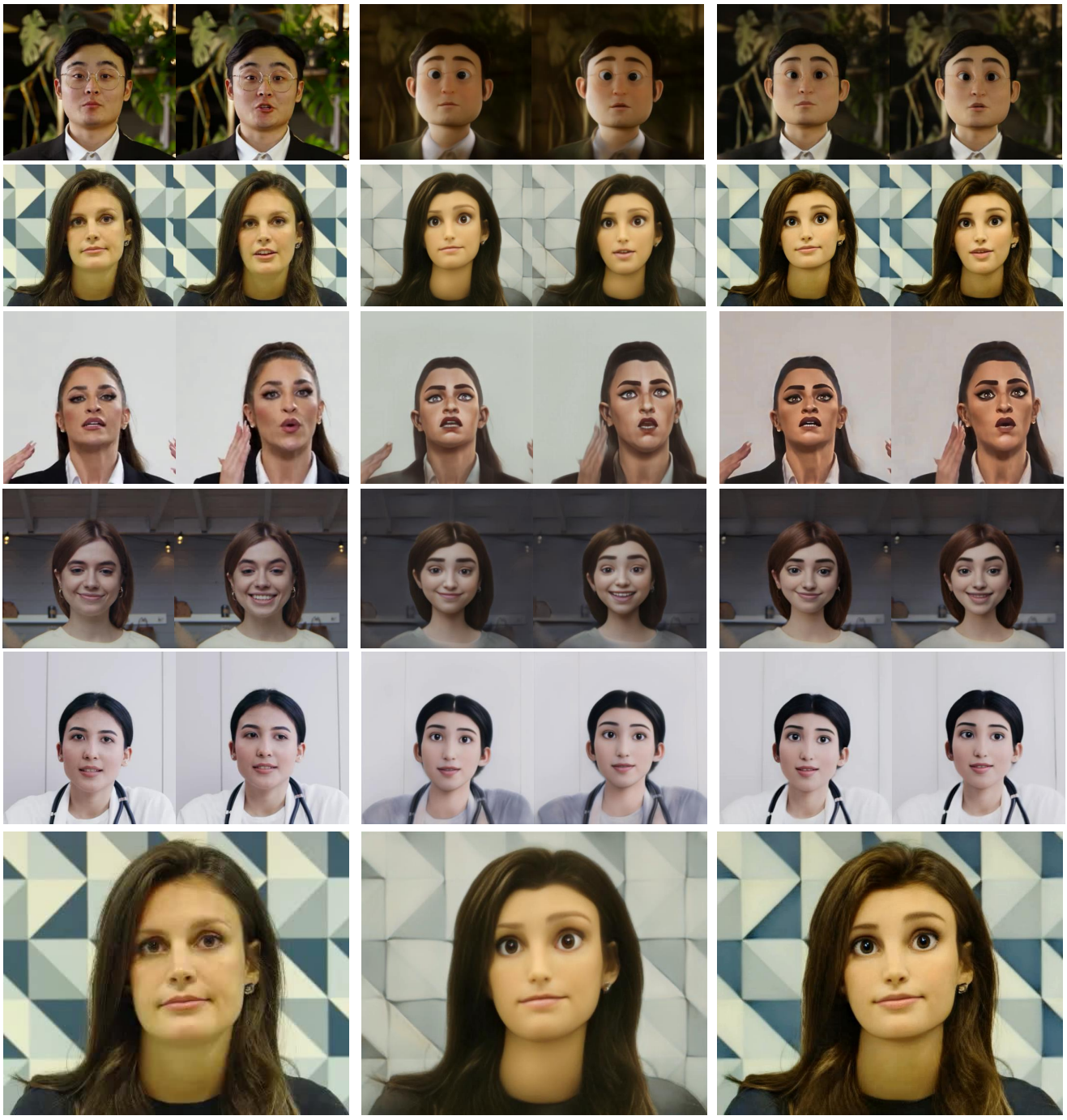



Figure 8. **Comparison on video toonify** (Part I).

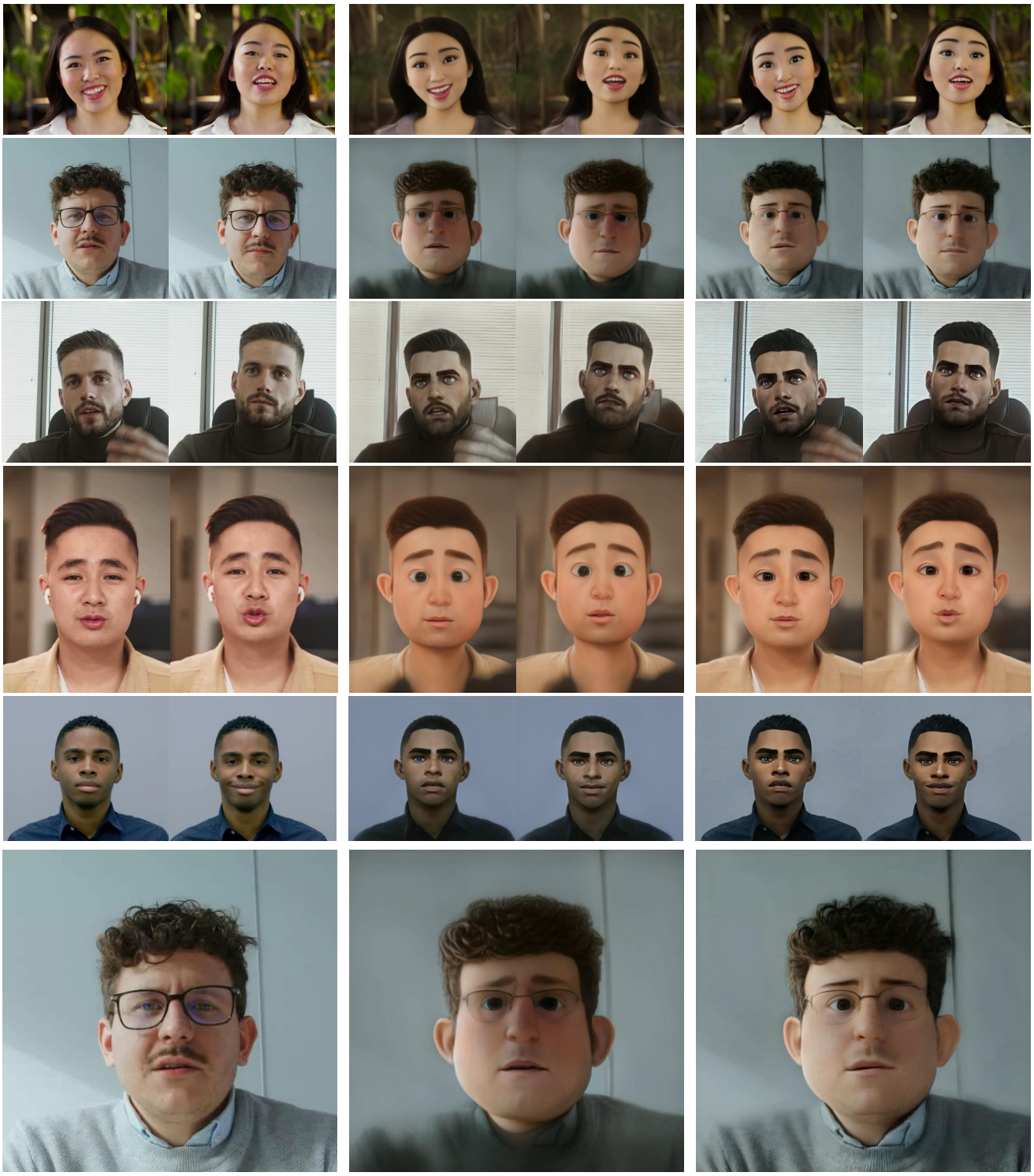



Figure 9. **Comparison on video toonify** (Part II).

## 2.2. Quantitative Evaluation

Besides the user studies on the tasks of sketch/mask-to-face translation and video toonify in the main paper, we present more quantitative comparisons on other tasks. Since our baselines are mainly designed for cropped aligned faces, their results either have unprocessed black regions or discontinuities along the seams between the original regions. In this case, it might be hard to find appropriate evaluation metrics from previous studies for rigorous comparison. Therefore, the quantitative scores in this section are just for reference of the performance and we did not include them in the main paper.

### 2.2.1 Normal FoV face inversion

We use the first frame of 796 videos from FaceForensics++ [7] as a testing set to evaluate the quality of StyleGAN inversion. Part of the examples are shown in Figs. 1-2. We compare the LPIPS distance, the mean absolute error (MAE) and the mean squared error (MSE) between the reconstructed image and the input image. The results are shown in Table 1. It can be seen that the unprocessed black regions in pSp and HyperStyle's results greatly harm their scores. By comparison, our encoder (Step I) achieves better scores. Our full two-step inversion obtains the best scores. For a fair comparison, we use 500 iterations for all optimizations in Step II.

Table 1. **Qualitative evaluation of inversion**. Best scores are marked in bold.

| Metric | LPIPS↓ | MAE↓ | MSE↓ |
|---|---|---|---|
| pSp [6] | 0.539 | 0.486 | 0.547 |
| HyperStyle [1] | 0.518 | 0.472 | 0.542 |
| only Step I | 0.385 | 0.130 | 0.039 |
| only Step II | 0.120 | 0.122 | 0.055 |
| ours | **0.086** | **0.057** | **0.012** |

### 2.2.2 Normal FoV face super-resolution

We use the first frame of 796 videos from FaceForensics++ [7] as a testing set to evaluate the quality of face super-resolution. Part of the examples are shown in Figs. 3-4. PSp pays attention to the realism of the face, but lacks fidelity to the inputs. By comparison, our results are more consistent with the input faces, thus obtaining better scores in LPIPS, MAE and PSNR.

Table 2. **Qualitative evaluation of super-resolution**. Best scores are marked in bold.

| Metric | LPIPS↓ | MAE↓ | PSNR↑ |
|---|---|---|---|
| pSp [6] + Real-ESRGAN [11] | 0.386 | 0.105 | 21.638 |
| ours | 0.356 | 0.084 | 24.257 |
| ours-32 | **0.304** | **0.068** | **25.617** |

### 2.2.3 Video face attribute editing

We use 28 videos from FaceForensics++ [7] as a testing set to evaluate the quality of face attribute editing. Part of the examples are shown in Fig. 7. For temporal consistency, we use ID-c and ID-m as metrics:

- Identity consistency (ID-c): It measures the consistency between the edited face and the input face. We calculate the identity loss [2] between each edited frame and the original frame.
- Identity maintenance (ID-m): It measures the preservation of the identity along all edited frames. For each edited video clip, we calculate the identity loss between the generated frames and the first edited frame.

Table 3 reports the averaged ID-c and ID-m over all the video clips and our method achieves the best temporal consistency in terms of identity consistency and maintenance.

For video quality, we use frechet video distance (FVD) [9] as the evaluation metric. We resize all videos to $224 \times 224$ and use the first $150$ frames of each video to calculate FVD. Table 3 reports the averaged FVD over two editing tasks, and our method obtains the highest video quality in both tasks.

Table 3. **Qualitative evaluation of video editing**. Best scores are marked in bold.

| Task | hair color editing | | | age editing | | | average | | |
|---|---|---|---|---|---|---|---|---|---|
| Metric | ID-c↓ | ID-m↓ | FVD↓ | ID-c↓ | ID-m↓ | FVD↓ | ID-c↓ | ID-m↓ | FVD↓ |
| pSp [6] | 0.174 | 0.324 | 1212.72 | 0.167 | 0.326 | 1277.92 | 0.171 | 0.325 | 1245.32 |
| HyperStyle [1] | 0.117 | 0.319 | 416.71 | 0.130 | 0.320 | 448.56 | 0.124 | 0.320 | 432.64 |
| ours | **0.048** | **0.312** | **186.31** | **0.055** | **0.308** | **210.38** | **0.052** | **0.310** | **198.35** |

## 2.3. Supplementary Domain Transfer Results

After StyleGANEX inversion, we can achieve full image style transfer by loading a new domain adapted StyleGAN model. Here, we show results on five different StyleGAN models provided by StyleGAN-NADA [3] in Fig. 10. Remarkably, our method successfully renders the full background with the target style, which cannot be simply achieved by cropping, editing and pasting.

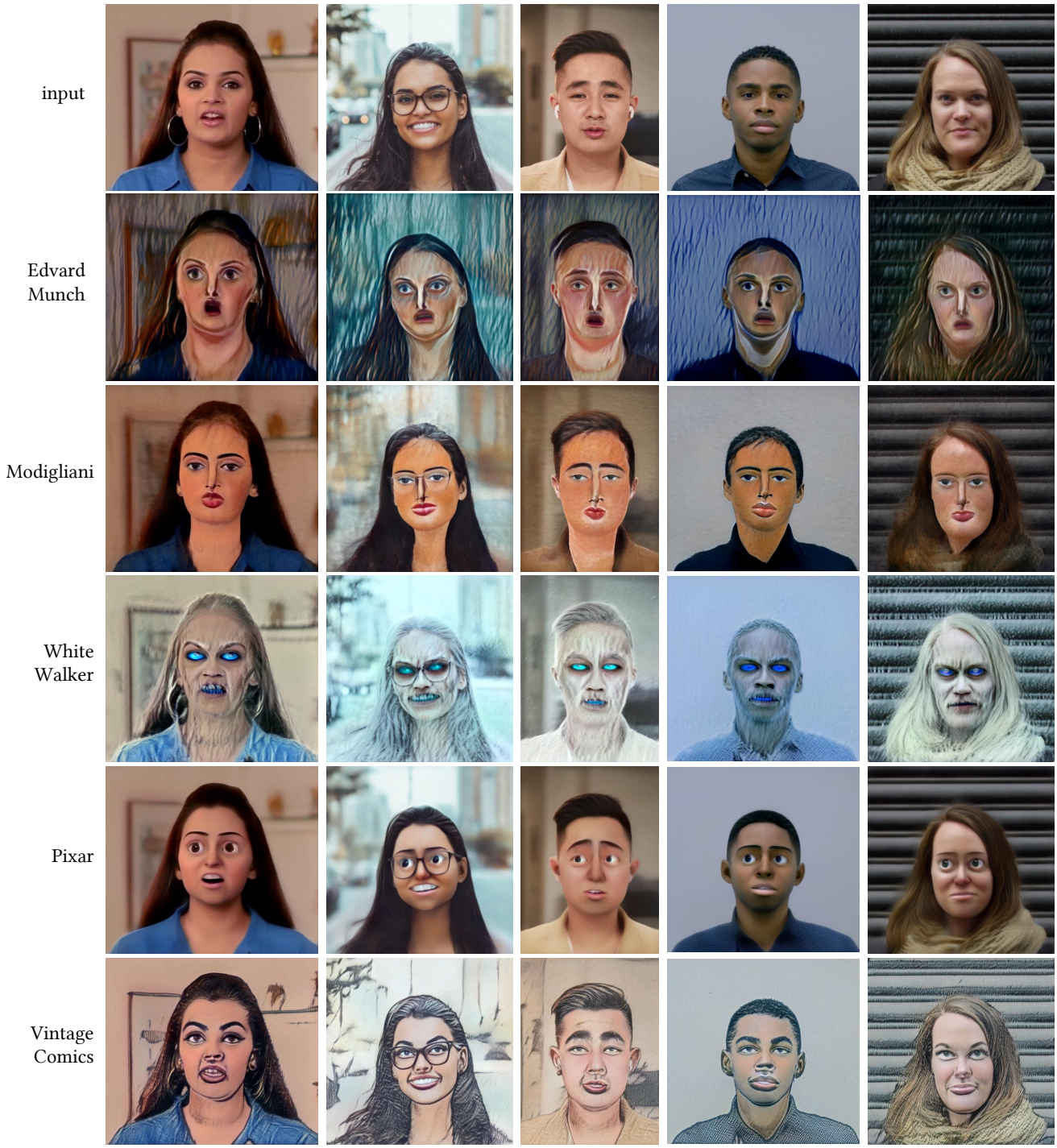



Figure 10. **Full image stylization results**.

## 3. Compatibility to StyleGAN

StyleGANEX is fully compatible with StyleGAN and can directly load a pre-trained StyleGAN model without training. In Fig. 11, we upsample the StyleGAN's constant input feature $f_0$ by $8\times$ with nearest neighbor interpolation to serve as the first-layer feature of StyleGANEX, StyleGANEX generates the same face image as the StyleGAN from the same latent code $w^+$. Formally, we have $G(f_{0\uparrow}, w^+) = G_0(w^+)$, where $G$ and $G_0$ are StyleGANEX and StyleGAN, respectively. $f_{0\uparrow}$ is the $8\times$ upsampled $f_0$ with nearest neighbor interpolation.

(a) StyleGAN (b) StyleGANEX

Figure 11. **StyleGANEX is compatible with StyleGAN**.

## References


[1] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 18511–18521, 2022. 2, 4, 12, 13

[2] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 13

[3] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022. 2, 14

[4] Liming Jiang, Changxu Zhang, Mingyang Huang, Chunxiao Liu, Jianping Shi, and Chen Change Loy. Tsit: A simple and versatile framework for image-to-image translation. In *Proc. European Conf. Computer Vision*, pages 206–222. Springer, 2020. 2, 7

[5] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 2

[6] 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 *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2021. 2, 4, 12, 13

[7] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In *Proc. Int'l Conf. Computer Vision*, 2019. 2, 12, 13

[8] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 9243–9252, 2020. 2

[9] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 13

[10] 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 *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2018. 2, 7, 8

[11] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proc. Int'l Conf. Computer Vision*, pages 1905–1914, 2021. 5, 12

[12] Shuai Yang, Liming Jiang, Ziwei Liu, and Chen Change Loy. Vtoonify: Controllable high-resolution portrait video style transfer. *ACM Transactions on Graphics*, 41(6):1–15, 2022. 2, 10

[13] Jiapeng Zhu, Ruili Feng, Yujun Shen, Deli Zhao, Zheng-Jun Zha, Jingren Zhou, and Qifeng Chen. Low-rank subspaces in gans. In *Advances in Neural Information Processing Systems*, volume 34, 2021. 2