Deformable One-shot Face Stylization via DINO Semantic Guidance Supplementary Materials

Yang Zhou Zichong Chen Hui Huang*
Visual Computing Research Center, Shenzhen University

This supplementary material mainly contains the implementation details of our method, the comparisons, and the user study we conducted. More experimental results generated by our method and the verification in feature selection of DINO semantic guidance are also included.

A. Implementation Details

A.1. Network architecture

We use the StyleGANv2 architecture[1] [7], pre-trained on FFHQ [6] as our source model for human face stylization. We also use StyleGAN2-ada[2] [5], pre-trained on AFHQ [2] for testing on animal face stylization (*e.g.*, cat and dog); See Sec. E.

For the STN blocks appended in the generator, we construct them following [4], where each contains three components: a localization net, a grid generator, and a sampler. The TPS-STN and Basic-STN share the same localization net architecture, two convolution layers with `MaxPool` operation, followed by two fully connected layers (FC). We use ReLU activation for both the CNN and linear layers. We set 5×5 kernel size for convolution layers and output channels to 128. We also set the output size of the two FC layers to 64 and 200[3], respectively.

A.2. Baseline methods

We compare with four state-of-the-art one-shot face stylization methods, which are MTG[4] [13], JoJoGAN[5] [3], DiFa[6] [11] and OneshotCLIP[7] [8]. We train the models by their released source codes with default settings.

A.3. Variants of MTG and JoJoGAN

For fairness, we convert MTG [13] and JoJoGAN [3], named MTG-pair and JoJoGAN-pair, to accept a paired reference for training. For convenience, we mark the original generator and the fine-tuned generator as G^s and G^t, respectively. The real-style paired reference is represented by (I^s_{ref}, I^t_{ref}).

MTG-pair. See [13] for details of MTG. Different from the original version, we obtain the reference cross-domain vector using the given paired reference directly, which does not require to find a latent code w^t_{ref} for an image similar to I^t_{ref} that is plausibly within source domain during the construction of v^{ref}. Thus, the v^{ref} is modified to

$$v^{ref} = E_I(I^t_{ref}) - E_I(I^s_{ref}), \tag{1}$$

where E_I denotes the CLIP image-embedding model.

Since it is important to match the style reference I^t_{ref} with the generated image $G^t(w^s_{ref})$, we additionally obtain w^s_{ref} by II2S [14]. Then, we take $G^t(w^s_{ref})$ into the calculation of $\mathcal{L}_{ref\_clip}$ and $\mathcal{L}_{ref\_rec}$ in MTG, and use the full losses of MTG for training. Besides, we use the same implementation as MTG for inference.

JoJoGAN-pair. See [3] for details of JoJoGAN. Similar to MTG-pair, we are not required to find a latent code w^t_{ref} of I^t_{ref}. Instead, we obtain the reference latent code w^s_{ref} by inverting I^s_{ref} using e4e [10] and use the code to create the training set by random style mixing. Following [3], we use the same perceptual loss calculated by features from the discriminator for training.

A.4. Training data

Fig. 1 shows the paired references used in our experiments. The paired references are obtained from the existing models, except for the last three columns collected from the internet.

A.5. Training time

We did all the experiments using a single NVIDIA RTX 3090 and recorded the training time. Table 1 shows the average training time over five times for our method and the

*Corresponding author.
[1] https://github.com/rosinality/stylegan2-pytorch
[2] https://github.com/NVlabs/stylegan2-ada-pytorch
[3] Here, 200 is because we set the grid size of STNs to 10×10.
[4] https://github.com/ZPdesu/MindTheGap
[5] https://github.com/mchong6/JoJoGAN
[6] https://github.com/YBYBZhang/DiFa
[7] https://github.com/cyclomon/OneshotCLIP

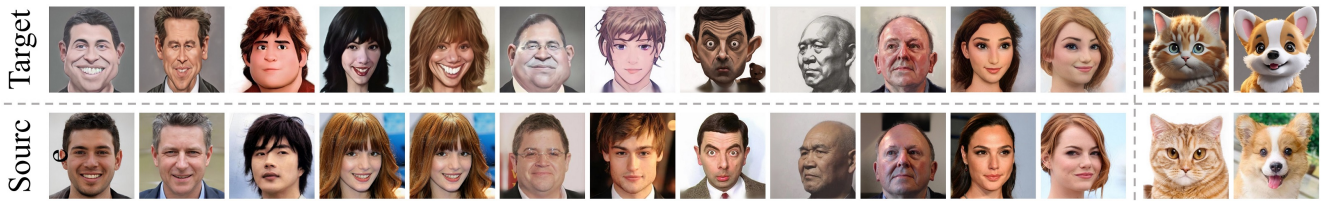

Figure 1. Paired training data.

competitors. Note that for MTG, JoJoGAN and our method, the time spent on reference inversion is not included. As a result, our method still maintains a short fine-tuning time.

Table 1. **Average training time** over 5 times of each.

Method	MTG [13]	JoJoGAN [3]	DiFa [11]
Avg. Time	8m50s	31s	2m2s
Method	OneshotCLIP [8]	Ours	
Avg. Time	49m48s	12m16s	

B. DINO Semantic Guidance

B.1. Feature selection

As the representations vary across different facets of ViT in different layers according to [1], we conducted a simple over-fitting test for selecting DINO features.

With the DINO semantic guidance, our objective is to deform real faces based on a style/deformation reference while keeping the global semantics unchanged. In the over-fitting test, we use a selected latent code w_* as a training sample and optimize the StyleGAN generator G with a directional deformation loss and a structure preservation loss. We employ the directional deformation loss $\mathcal{L}_{direct}$ as outlined in the main paper. Additionally, we compute the structure preservation loss $\mathcal{L}_{struct}$ using DINO features of $G(w_*)$ and I_{w_*}, represented as follows:

$$\mathcal{L}_{struct} = ||E_D^l(G(w_*)) - E_D^l(I_{w_*})||_2^2, \qquad (2)$$

where $E_D^l(\cdot)$ denotes the l-th layer of DINO features, $I_{w_*} = G^s(w_*)$ is the image generated by the original generator G^s.

Fig. 2 shows the training losses recorded in our over-fitting test. As illustrated in Fig. 2(a) and (b), when utilizing the Tokens of DINO, we achieve a closer match to directional guidance while simultaneously preserving the structure, outperforming the Keys. Consequently, we opt to utilize Tokens as the feature representation in our framework. Moreover, in our pursuit of identifying more suitable feature layers for representation, we experiment with various layer combinations. Specifically, we examine the features of the 6th and 12th layer for the computation of the two losses. As depicted in Figure 2(c) and (d), using Tokens from the 6th layer for $\mathcal{L}_{struct}$ and Tokens from either the 6th layer or

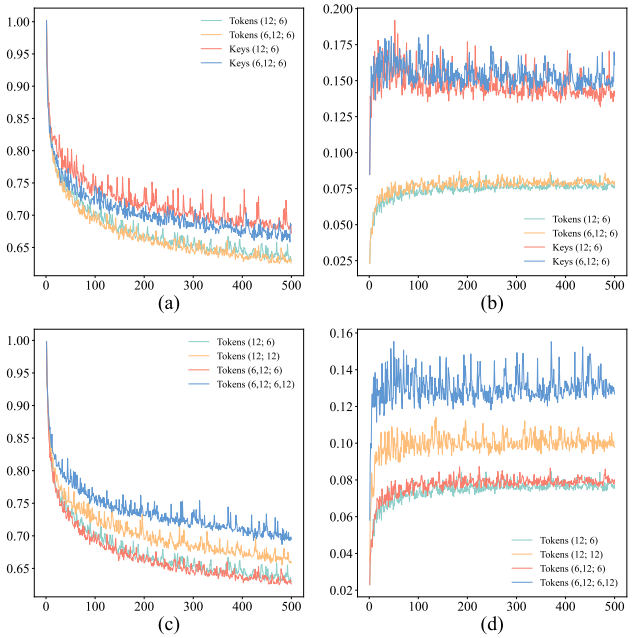

Figure 2. Visualization of training loss curves of different DINO features. (a) and (c) are the curves of $\mathcal{L}_{direct}$ while (b) and (d) are the curves of $\mathcal{L}_{struct}$. We explore the representations of Tokens and Keys in (a) and (b), and explore the representations of different layers of Tokens in (c) and (d). The legend tags denote the combination of different layers of Tokens/Keys. The feature layer(s) used in $\mathcal{L}_{direct}$ and $\mathcal{L}_{struct}$ are presented in left and right of the tuple, respectively.

a combination of the 6th and 12th layers for $\mathcal{L}_{direct}$ yields a superior representation. Given that the combination of `Tokens (6,12; 6)` slightly outperforms `Tokens (12; 6)` in matching the directional reference, we incorporate a mixture of M- and H-level features of DINO for the computing the directional deformation guidance. Additionally, we exclusively utilize M-level features to ensure relative structural consistency in our framework.

B.2. Comparison with weakly-supervised ViTs

In addition to the PCA-based hierarchical feature visualization shown in our submitted paper, we further present an experimental result to prove the DINO features are better than those of existing weakly-supervised ViTs (*i.e.*, CLIP [9] and FaRL [12]) for semantic representation in the

Figure 3. Comparison with weakly-supervised ViTs as the feature representation.

task of face stylization. We use the same configuration of our framework and simply replace the DINO features with those from CLIP and FaRL. Fig. 3 shows the comparison using different ViTs as feature representations. Apparently, DINO features surpass the CLIP and FaRL both in geometry deformation and color style.

C. Effect of Color Alignment

Fig. 11 shows the generated training samples after color alignment. Our goal is to align the color of generated training samples to the reference, further ensuring the correctness of DINO semantic guidance. Thanks to the style mixing of StyleGAN generator, we can implement color alignment by swapping the latent codes in W+ space. Note that the fine-level codes in W+ space of StyleGAN mainly control the appearance of images, specifically for the 9th code. Therefore, we decided to swap the 9-18th codes of w with the corresponding codes of w^s_{ref} and w^t_{ref}. Moreover, We only use the color-aligned images to compute our two DINO-based losses, while we use the original images to compute the adversarial style loss.

D. Details of User Study

Fig. 8 shows the interface of our user study system. In our user study, we select five artistic styles to investigate human evaluation. Fig. 9 illustrates the detailed user preference for each style.

E. Deformable Face Stylization Results

E.1. Qualitative comparison with the SOTA

In addition to the examples shown in the main paper, we show more visual comparison results in Fig. 10 with MTG [13], JoJoGAN [3], DiFa [11], OneshotCLIP [8] and the variants of MTG and JoJoGAN. Besides the cases with strong exaggeration, we also test our approach on styles with less exaggeration, where in these cases the style references show more faithful identity correlation with the paired natural facial images. As shown in Fig. 4, our approach faithfully preserves the input face ID, while still vividly capturing both the appearance style and local deformations. As a comparison, JoJoGAN is less appealing in mimicking the style and appearance change.

Table 2. Comparison with SOTA on Inception Score (IS) (↑), where the three styles referred to below are those involved in Figures 7 and 8 of our main paper.

	MTG	JoJoGAN	MTG-pair	JoJoGAN-pair	Ours
Style1	1.43	1.39	1.32	1.38	**1.55**
Style2	1.20	1.29	1.32	1.33	**1.36**
Style3	1.51	1.52	1.22	1.36	**1.62**

Furthermore, we show the compatibility of our approach to unpaired cases. As shown in Fig. 5, due to the unfaithful mapping of cross-domain GAN inversion, the signature exaggeration exhibited in the style example will not be retained in the results of such unpaired cases.

E.2. Quantitative comparison with the SOTA

In the main paper, we have evaluated the generated results from three aspects: perception, deformation, and face identity. Recognizing the significance of addressing the mode collapse issue in few-shot learning, we further calculate the Inception Score (IS) to verify the generation diversity of our approach. As listed in Tab. 2, our method achieves better generation diversity over SOTA methods.

E.3. Arbitrary artistic portrait generation

Fig. 12-19 show artistic portraits with different styles generated by our fine-tuned models.

E.4. Deformable face stylization on animal domains

We also test our method for animal face stylization. We use the StyleGAN2-ada [5] pre-trained on AFHQ [2] (cat and dog) as the base generator. We form the deformation-aware generator by inserting the STNs into the base generator. Note that we use the same network settings, apart from the weight of relative structural consistency loss, where we set it to $1e4$. Fig. 7 shows the randomly generated results on cat and dog faces. More random generated results are shown in Fig. 20 and Fig. 21.

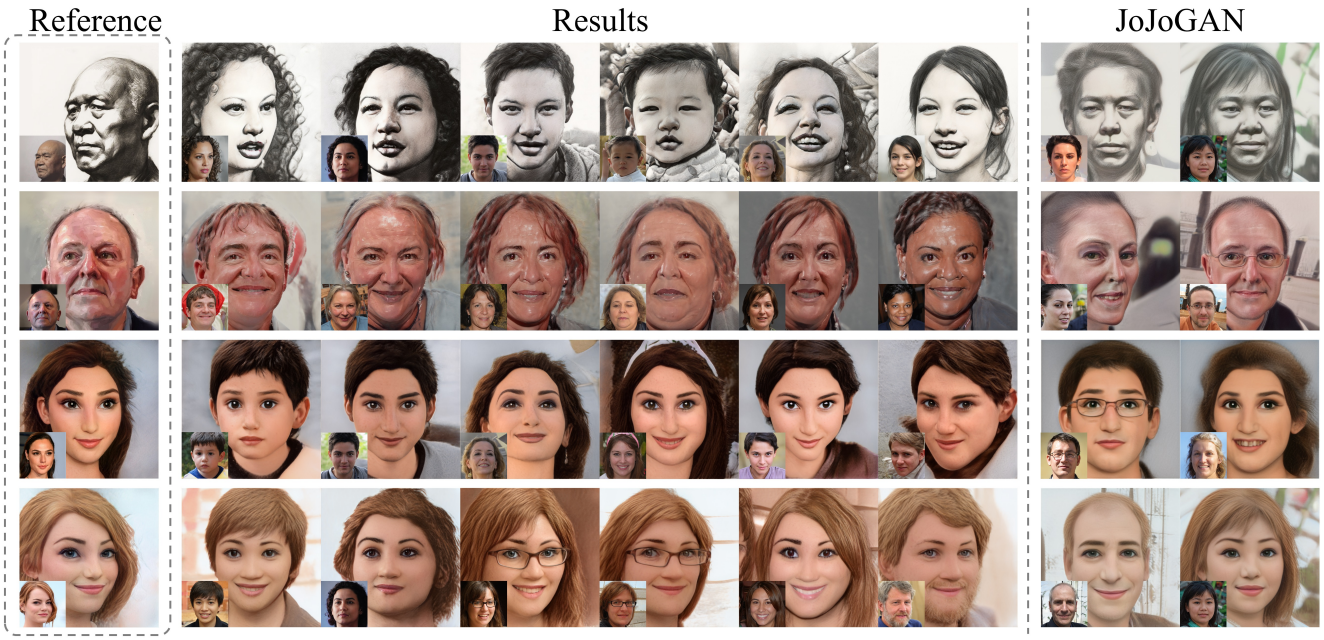

Figure 4. Results on styles with less exaggeration and more faithful identity correlations between the paired references. Our method still surpasses JoJoGAN in capturing both the appearance style and local deformations of the style examples; *e.g.*, the bright spot on the faces in row 2, and the curly hairstyle in row 4.

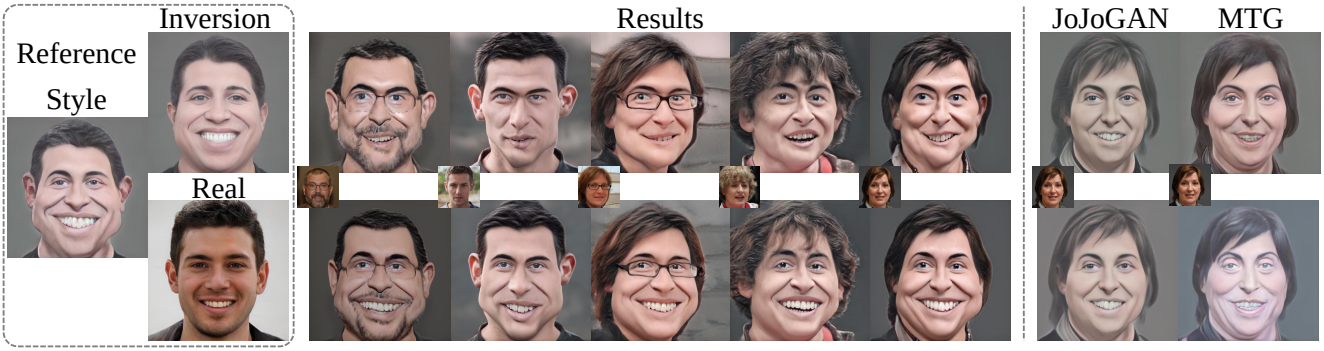

Figure 5. Results using unpaired data (top). Compared with using paired data (bottom), the signature smile of the style image is lost.

E.5. Controllable face deformation

Fig. 6 shows the additional controllable face deformation in different styles. Taking the trained STNs as plug-ins, we provide the controllability for face deformation, allowing us to flexibly change the deformation degree of faces.

References

[1] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *ECCVW What is Motion For?*, 2022. 2

[2] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*, 2020. 1, 3

[3] Min Jin Chong and David Forsyth. Jojogan: One shot face stylization. In *Proc. of Euro. Conf. on Computer Vision*, 2022. 1, 2, 3

[4] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 1

[5] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems*, 2020. 1, 3

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

Figure 6. Controllable deformation.

Figure 7. Deformable face stylization on animal faces.

[7] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*, 2020. 1

[8] Gihyun Kwon and Jong Chul Ye. One-shot adaptation of gan in just one clip. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 45(10):12179–12191, 2023. 1, 2, 3

[9] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 2

[10] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *arXiv preprint arXiv:2102.02766*, 2021. 1

[11] Yabo Zhang, mingshuai Yao, Yuxiang Wei, Zhilong Ji, Jinfeng Bai, and Wangmeng Zuo. Towards diverse and faithful one-shot adaption of generative adversarial networks. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 1, 2, 3

[12] Yinglin Zheng, Hao Yang, Ting Zhang, Jianmin Bao, Dongdong Chen, Yangyu Huang, Lu Yuan, Dong Chen, Ming Zeng, and Fang Wen. General facial representation learning in a visual-linguistic manner. In *Proc. of IEEE Conf. on Computer Vision & Pattern Recognition*, pages 18676–18688, New Orleans, LA, USA, June 2022. IEEE. 2

[13] Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Mind the gap: Domain gap control for single shot domain adaptation for generative adversarial networks. In *Proc. of Int. Conf. on Learning Representations*, 2022. 1, 2, 3

[14] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Improved stylegan embedding: Where are the good latents? *CoRR*, abs/2012.09036, 2020. 1

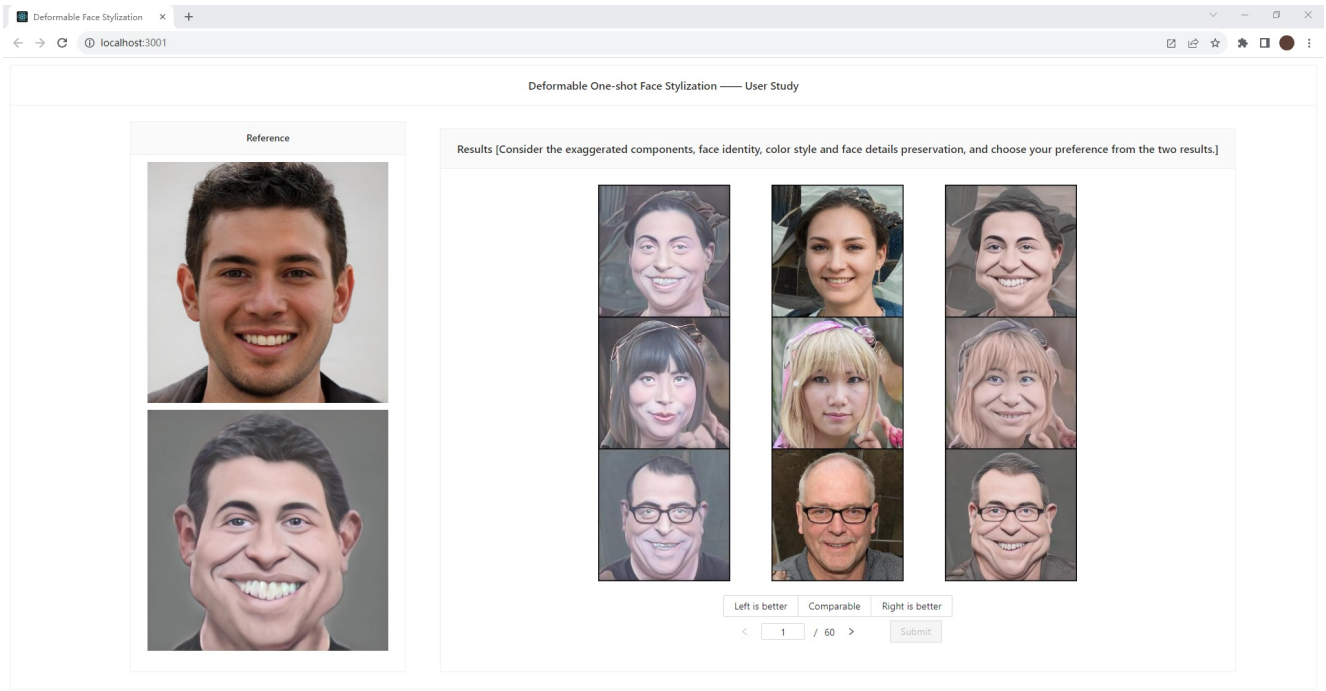

Figure 8. User interface.

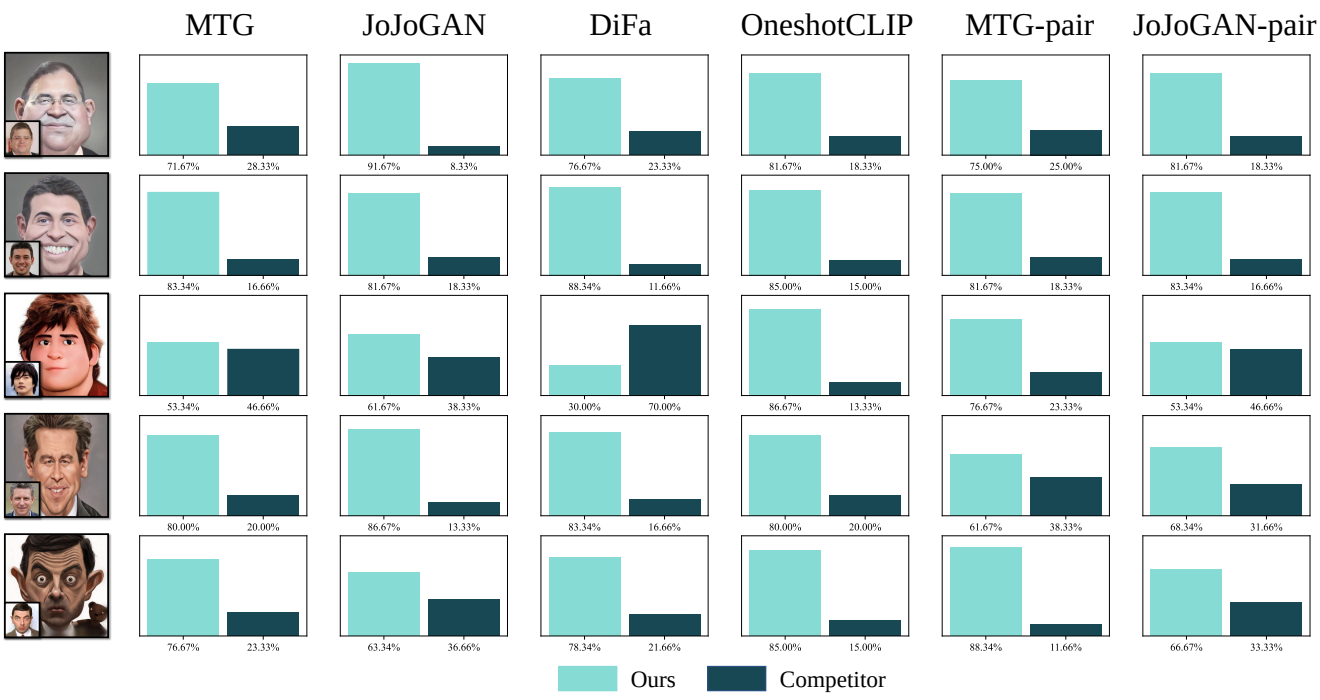

Figure 9. User preference.

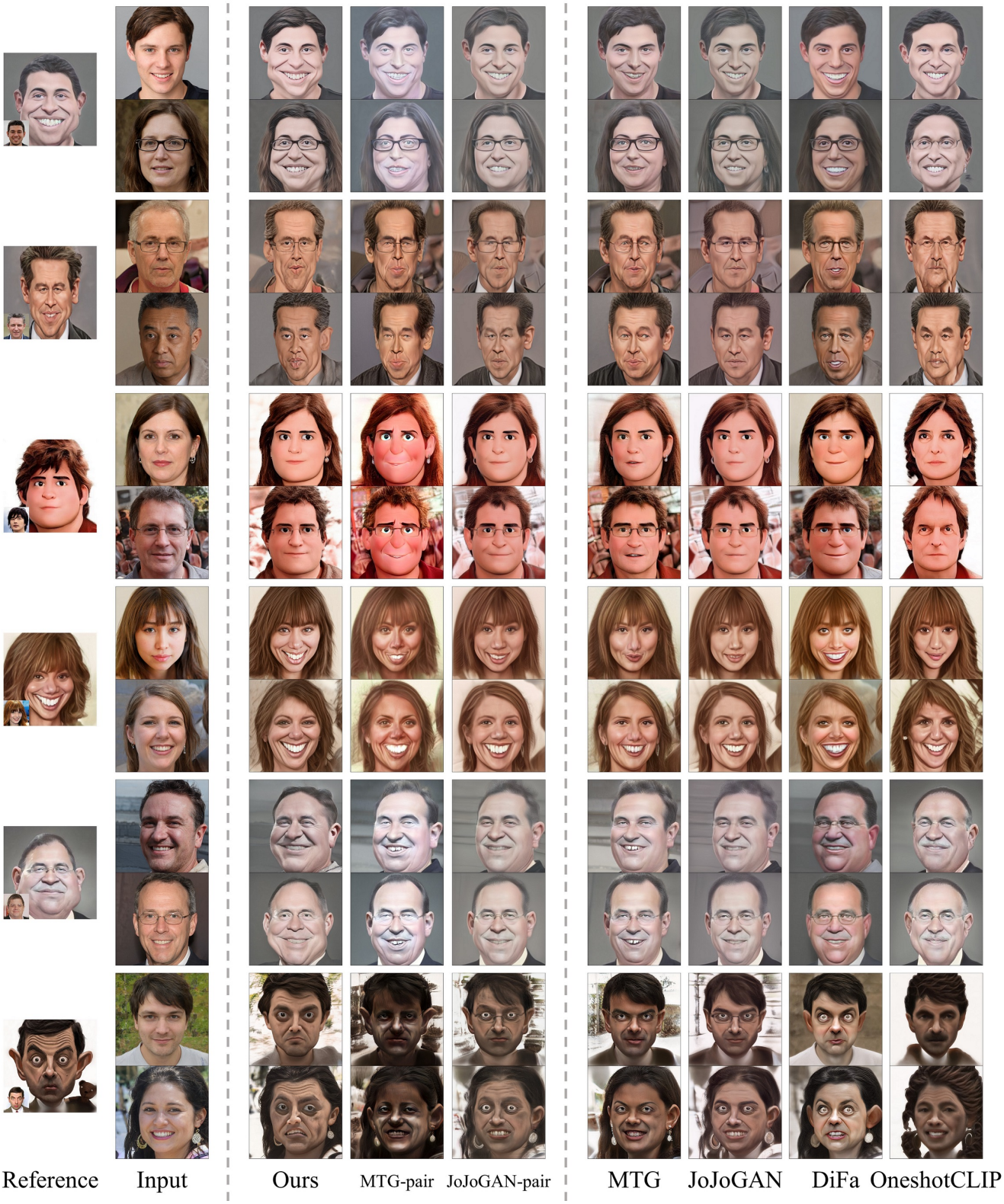

Figure 10. Visual comparison on different face style transfer.

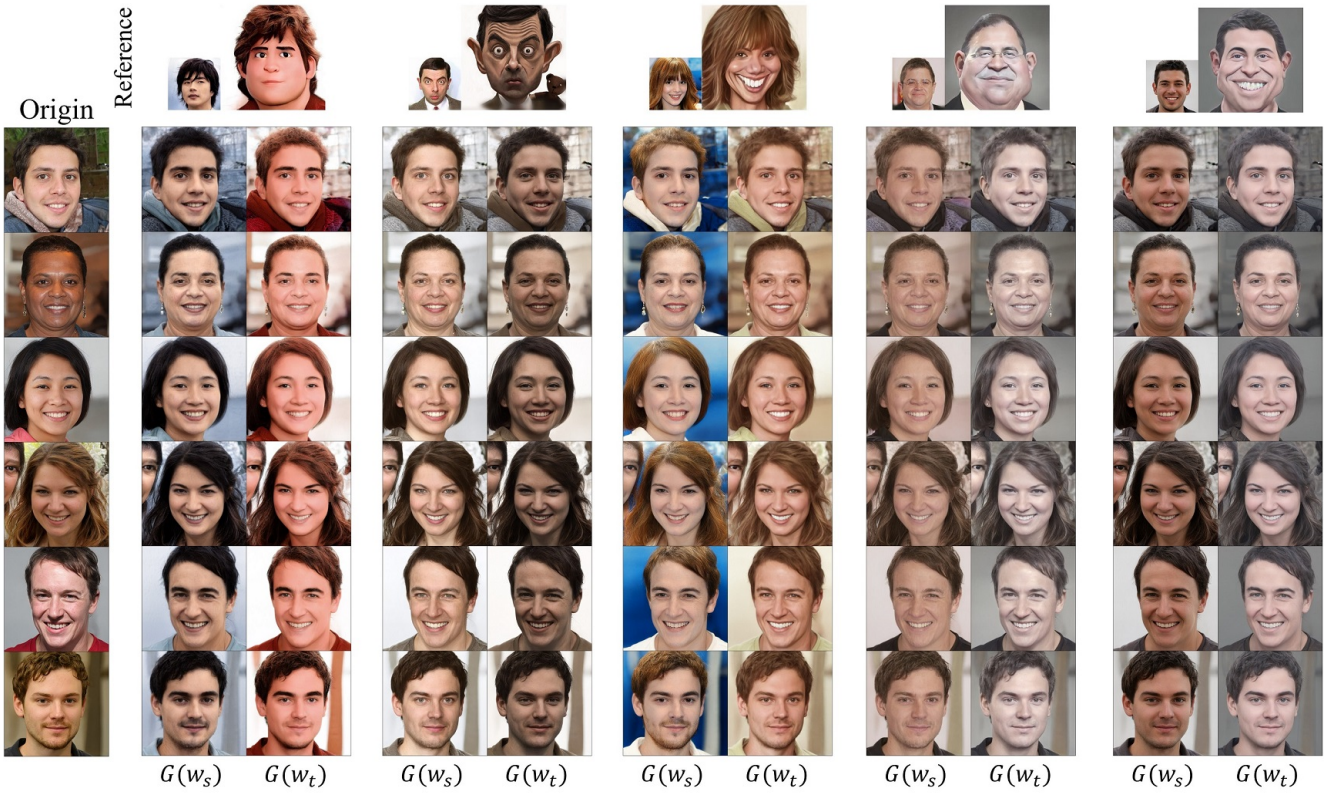

Figure 11. Color alignment by style mixing.

Figure 12. Random generated results of face stylization using paired reference shown at left bottom.

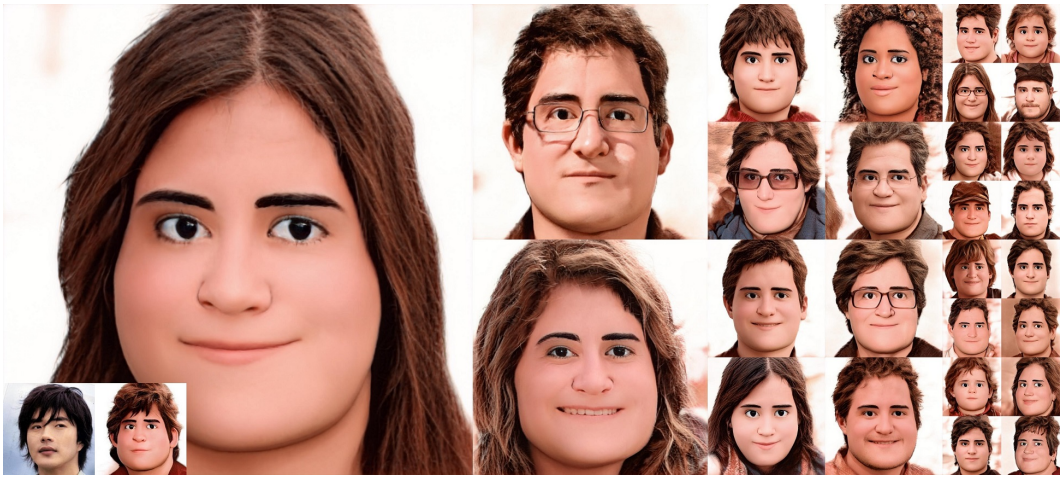

Figure 13. Random generated results of face stylization using paired reference shown at left bottom.

Figure 14. Random generated results of face stylization using paired reference shown at left bottom.

Figure 15. Random generated results of face stylization using paired reference shown at left bottom.

Figure 16. Random generated results of face stylization using paired reference shown at left bottom.

Figure 17. Random generated results of face stylization using paired reference shown at left bottom.

Figure 18. Random generated results of face stylization using paired reference shown at left bottom.

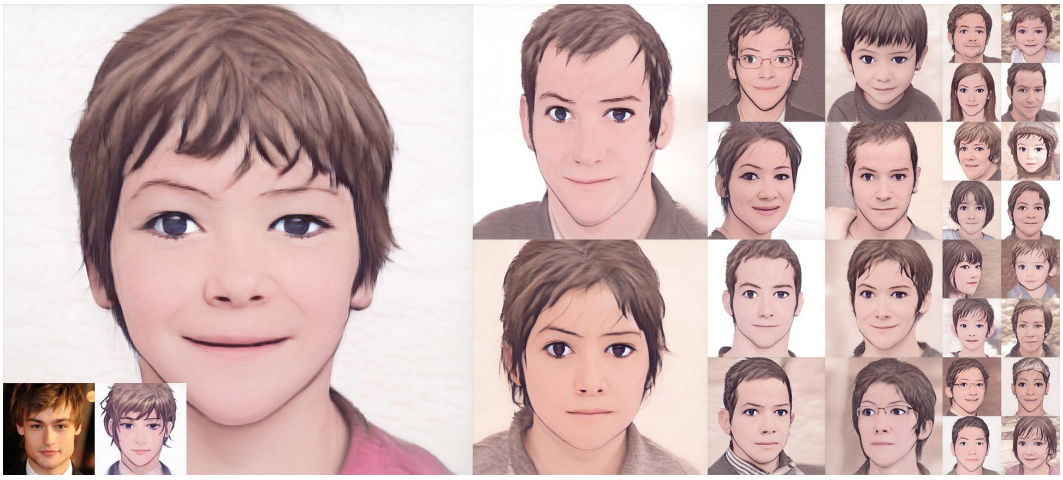

Figure 19. Random generated results of face stylization using paired reference shown at left bottom.

Figure 20. Deformable face stylization on cat faces. Paired reference is shown at left bottom.

Figure 21. Deformable face stylization on dog faces. Paired reference is shown at left bottom.