One-Shot Structure-Aware Stylized Image Synthesis

– Supplementary Material –

A. Training Details

A.1. Training Preparation

In order to train one-shot structure-aware stylized image synthesis (OSASIS), it is necessary to create a photorealistic image, denoted as I_A^{style}, that is semantically aligned with a given style image, I_B^{style}. This is achieved by first encoding I_B^{style} into a latent code, represented as $\mathbf{x}_{\mathbf{t}_0}$ using Eq.9. Then I_A^{style} is generated from $\mathbf{x}_{\mathbf{t}_0}$ using Eq.10, utilizing a pretrained DDPM ϵ_θ during its encoding and generation phases. However, since these processes are stochastic in nature, the resulting I_A^{style} may not always align perfectly with I_B^{style}. To overcome this, we generate 30 images from I_B^{style} and evaluate their alignment with I_B^{style}using both the L_1 loss and perceptual similarity loss [10]. We then select the image that is most similar to I_B^{style} as I_A^{style} as the prime candidate. In cases of out-of-domain (OOD) reference images, we match the domain of the pretrained DDPM with the domain of the style image. (*e.g.* for church style images, we use a pretrained DDPM trained on LSUN-church).

A.2. Structure-Preserving Network

The structure-preserving network (SPN) incorporates a 1x1 convolution to preserve the overall structure of the input image. During the reverse process, the SPN's output is integrated with the DDIM output. Given that this integration takes place at every timestep, it is crucial for the network to recognize the timestep to effectively control structure preservation. To facilitate this, each block of the SPN is conditioned on the timestep. The detailed architecture of the SPN is illustrated in Figure 10.

A.3. Loss Function Formulation

Cross-Domain Loss The objective of the cross-domain loss is to align the directional shifts from domain A to domain B, ensuring that the change from I_A^{in} to I_B^{in} is kept consistent with the change from I_A^{style} to I_B^{style}. Leveraging the CLIP image encoder E_I, latent vectors of both the input and style images are extracted. Subtracting these semantically aligned latent vectors results in semantically meaningful directions. The changes in the style and input images are calculated using Eq.15 and Eq.16, respectively. The cosine similarity, as described in Eq. 17, is then used to evaluate the similarity of the two directions.

$$\mathbf{v}_{\text{style}} = E_I(I_B^{\text{style}}) - E_I(I_A^{\text{style}}) \tag{15}$$

$$\mathbf{v}_{\text{in}} = E_I(I_B^{\text{in}}) - E_I(I_A^{\text{in}}) \tag{16}$$

$$L_{cross} = 1 - sim(\mathbf{v}_{style}, \mathbf{v}_{in}) \tag{17}$$

In-Domain Loss The purpose of the in-domain loss is to mitigate unintended changes in the direction of stylization, which can often result in excessive reflection of the style image. This is achieved by measuring the similarity of changes within both domains A and B. Like the cross-domain loss, the in-domain changes are calculated using Eq.18 and Eq.19. The similarity between the two directions is then determined using Eq. 20.

$$\mathbf{v}_A = E_I(I_A^{\text{cont}}) - E_I(I_A^{\text{style}}) \tag{18}$$

$$\mathbf{v}_B = E_I(I_B^{\text{cont}}) - E_I(I_B^{\text{style}}) \tag{19}$$

$$L_{In} = 1 - sim(\mathbf{v}_A, \mathbf{v}_B) \tag{20}$$

Reconstruction Loss The reconstruction loss aims to guarantee that the photorealistic style image I_A^{style} can be accurately translated from domain A to domain B. This is achieved by encoding I_A^{style} using $\boldsymbol{\epsilon}_\theta^A$, which results in the latent vectors $\mathbf{z}_{\text{sem}}^{\text{style}}$ and $\mathbf{x}_{\mathbf{t}_0}^{\text{style}}$. The latent vectors are then fed into $\boldsymbol{\epsilon}_\theta^B$, which generates the predicted domain B style image $\hat{I}_B^{\text{style}}$. The reconstruction loss is calculated by comparing $\hat{I}_B^{\text{style}}$ with I_B^{style} and the summation of the L_1 loss, perceptual similarity loss [10], and the L_1 CLIP embedding loss described in Eq. 22-24. The total reconstruct loss can be computed using Eq. 25.

$$\hat{I_B^{\text{style}}} = \epsilon_\theta^B(\mathbf{z}_{\text{sem}}^{\text{style}}, \mathbf{x}_T^{\text{style}}) \tag{21}$$

$$L_{re-image} = L_1(I_B^{\text{style}}, \hat{I_B^{\text{style}}}) \tag{22}$$

$$L_{re-lpips} = L_{\text{lpips}}(I_B^{\text{style}}, \hat{I_B^{\text{style}}}) \tag{23}$$

$$L_{re-clip} = L_1(E_I(I_B^{\text{style}}), E_I(\hat{I_B^{\text{style}}})) \tag{24}$$

$$L_{recon} = \lambda_{re-image}L_{re-image} + \lambda_{re-lpips}L_{re-lpips} + \lambda_{re-clip}L_{re-clip} \tag{25}$$

Total Loss The total loss is a weighted sum of the aforementioned cross-domain, in-domain, and reconstruction loss, formulated by the following equation:

$$L_{total} = \lambda_{cross}L_{cross} + \lambda_{in}L_{in} + L_{recon} \tag{26}$$

B. Sampling Details

B.1. Mixing Content and Style

After training the DDIM $\boldsymbol{\epsilon}_\theta^B$, we can combine the content of the input images with the style of style images. We achieve this by encoding these images into semantic latent codes, specifically $\mathbf{z}_{\text{sem}}^{\text{in}}$ for input and $\mathbf{z}_{\text{sem}}^{\text{style}}$for style. It is important to note that during training, $\mathbf{z}_{\text{sem}}^{\text{style}}$ comes from I_A^{style},

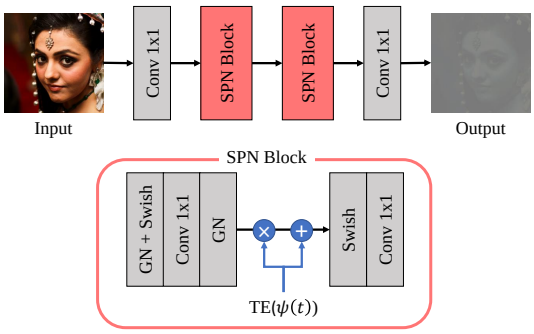

Figure 10. The SPN architecture is comprised of SPN blocks, each consisting of two 1x1 convolutions with group normalizations [9] and swish activations [5]. To incorporate the temporal information of each timestep, the SPN block employs sinusoidal position embeddings [7], represented as ψ for each time step. Timestep embedding (TE) layers are also incorporated, consisting of two linear layers. To condition the SPN block on the current timestep, we use the timestep embedding as a scale and shift parameter of group normalization, which is similar to previous works [3].

but for sampling, it is sourced from I_B^{style}. Using the pretrained DDIM $\boldsymbol{\epsilon}_\theta^A$, we encode I_A^{in} into a structural latent code $\mathbf{x}_{\mathbf{t}_0}^{\text{in}}$. From $\mathbf{x}_{\mathbf{t}_0}^{\text{in}}$, $\boldsymbol{\epsilon}_\theta^B$ generates a stylized image. The process of generating a stylized image is similar to the process of generating I_B^{in} from I_A^{in}, as described in Eq.12-14. However, the sampling step involves two semantic latent codes, $\mathbf{z}_{\text{sem}}^{\text{in}}$ and $\mathbf{z}_{\text{sem}}^{\text{style}}$, so the generation process is adjusted accordingly:

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}} f_\theta(\mathbf{x}'_t, t, \mathbf{z}_{\text{sem}}^{\text{in}}, \mathbf{z}_{\text{sem}}^{\text{style}}) + \sqrt{1-\alpha_{t-1}} \epsilon_\theta^B(\mathbf{x}'_t, t, \mathbf{z}_{\text{sem}}^{\text{in}}, \mathbf{z}_{\text{sem}}^{\text{style}}) \quad (27)$$

$\mathbf{z}_{\text{sem}}^{\text{style}}$ is conditioned on low-level feature maps, while $\mathbf{z}_{\text{sem}}^{\text{in}}$ is conditioned on high-level feature maps. The separation between these feature maps is done using f_{ch}. The overall sampling process is illustrated in Figure 12.

B.2. Text-driven Manipulation

To enable text-driven image manipulation, we utilize CLIP directional loss, as in previous works [4]. The CLIP directional loss aligns the source-to-target text change with the source-to-target image change, similar to the cross-domain loss. The CLIP text encoder is denoted as E_T, and the target and source text is represented as T_{trg} and T_{src}, respectively. The source image I_A^{in} is encoded into semantic and structural latent codes, $\mathbf{z}_{\text{sem}}^{\text{in}}$ and $\mathbf{x}_{\mathbf{t}_0}^{\text{in}}$, using DDIM $\boldsymbol{\epsilon}_\theta^A$. We freeze $\mathbf{x}_{\mathbf{t}_0}^{\text{in}}$ and $\boldsymbol{\epsilon}_\theta^A$, and optimize $\mathbf{z}_{\text{sem}}^{\text{in}}$ to obtain the optimal semantic latent code $\mathbf{z}_{\text{sem}}^{\text{in}*}$. Following this, employing $\mathbf{x}_{\mathbf{t}_0}^{\text{in}}$ and $\mathbf{z}_{\text{sem}}^{\text{in}*}$ allows us to derive I_{opt}^{in} utilizing DDIM $\boldsymbol{\epsilon}_\theta^B$. The CLIP directional loss is computed as follows:

$$\mathbf{v}_{\text{text}} = E_T(T_{\text{trg}}) - E_T(T_{\text{src}}) \quad (28)$$

$$\mathbf{v}_{\text{image}} = E_I(I_{opt}^{\text{in}}) - E_I(I_A^{\text{in}}) \quad (29)$$

$$L_{text} = 1 - sim(\mathbf{v}_{text}, \mathbf{v}_{image}) \quad (30)$$

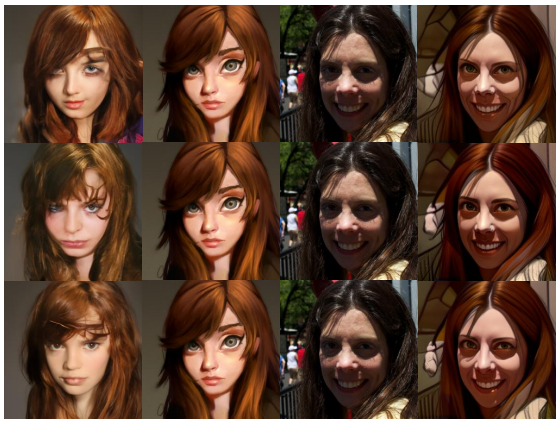

Figure 11. Reliance on I_A^{style}. The representation of I_A^{style} is stochastic, it can be adjusted through the sampling process transitioning from I_B^{style} to I_A^{style}. Notably, stylization results maintain consistent visual quality irrespective of variations in I_A^{style}.

C. Additional Experiments and Limitations

C.1. Experiments Settings

The encoding timestep t_0 is set to 500, equating to half of the total timesteps. Based on empirical findings, we define the loss function parameters as $\lambda_{re-image} = 10$, $\lambda_{re-lpips} = 10$, $\lambda_{re-clip} = 30$, $\lambda_{cross} = 1$, and $\lambda_{in} = 0.5$. For the SPN, we assign $\lambda_{SPN} = 0.1$. During sampling, f_{ch} is configured to 32, indicating that $\mathbf{z}_{\text{sem}}^{\text{style}}$ conditions up to the 32-resolution feature maps while other blocks are conditioned on $\mathbf{z}_{\text{sem}}^{\text{in}}$.

C.2. Reliance on I_A^{style}

The generation of I_A^{style} is inherently stochastic, leading to variations in its visual quality. However, the efficacy of our method is not intrinsically tied to the visual fidelity of I_A^{style}. As depicted in Figure 11, despite the varying visual presentations of I_A^{style}, our method consistently produces reliable stylization results. The resilience of our approach stems from the stylization trajectory determined in the CLIP space, thereby decoupling it from the aesthetic variations of I_A^{style}. To mitigate any misalignment that might arise between I_A^{style} and I_B^{style}, we adopt a systematic sampling methodology, subsequently auto-selecting the most congruent image, as outlined in **Training Preparation**.

C.3. Additional Results

In this section, we present additional stylization results and its comparisons with other stylization methods. Figure 13 shows the stylization outcomes of the original MTG [11] and JoJoGAN [2], which utilizes e4e [6] and ReStyle [1] respectively for inversion. Compared to HFGI [8], these methods struggle to preserve the structure of the input image leading to a loss of key elements, such as hands and

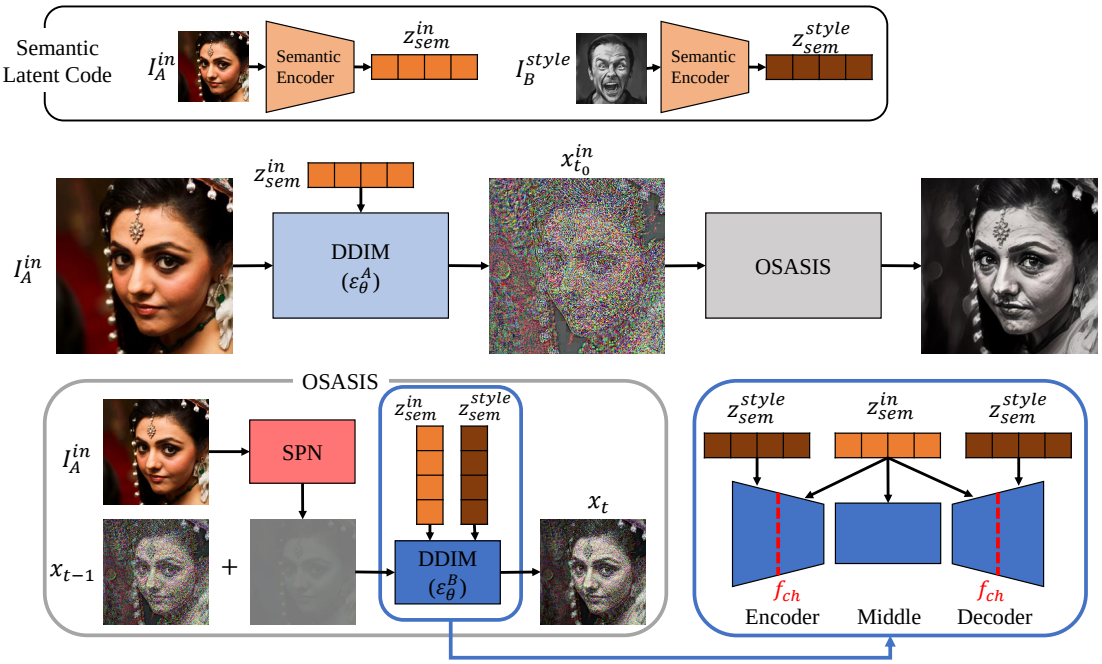

Figure 12. Sampling process of OSASIS. In contrast to the training phase, during the sampling process, $\mathbf{z}_{\text{sem}}^{\text{style}}$ is derived from I_B^{style}. To integrate the content from the input with the style of the style image, we condition the DDIM $\boldsymbol{\epsilon}_\theta^B$, on $\mathbf{z}_{\text{sem}}^{\text{style}}$ and $\mathbf{z}_{\text{sem}}^{\text{in}}$, respectively. This conditioning is separated by f_{ch}.

MTG	JoJoGAN	DiffuseIT	InST	OSASIS(Ours)
10.0%	20.0%	0.0%	5.0%	**65.0%**

Table 3. User study of stylized images from OSASIS and baselines

accessories. Additional stylization results on comparison methods are shown in Figure 14, where OSASIS outperforms other methods in structural preservation while stylizing. Figure 15 shows the stylization results of OOD reference images. Table 4 encapsulates a comprehensive quantitative comparison, encompassing both low and high-density image results.

C.4. User Evaluation results of qualitative samples

While quantitative assessments provide quality metrics, user studies offer deeper insights into a stylization model's effectiveness. Thus, we included the results of a preference-based user study in Table 3, where participants evaluated 20 stylization outcomes from OSASIS and its baselines against input and style images. The study documented the selection ratio for each method, aiming to discern the model that most effectively harmonizes input structure with stylistic elements. OSASIS emerged as the favored choice in Table 3, underscoring its excellence in meeting human perceptual standards.

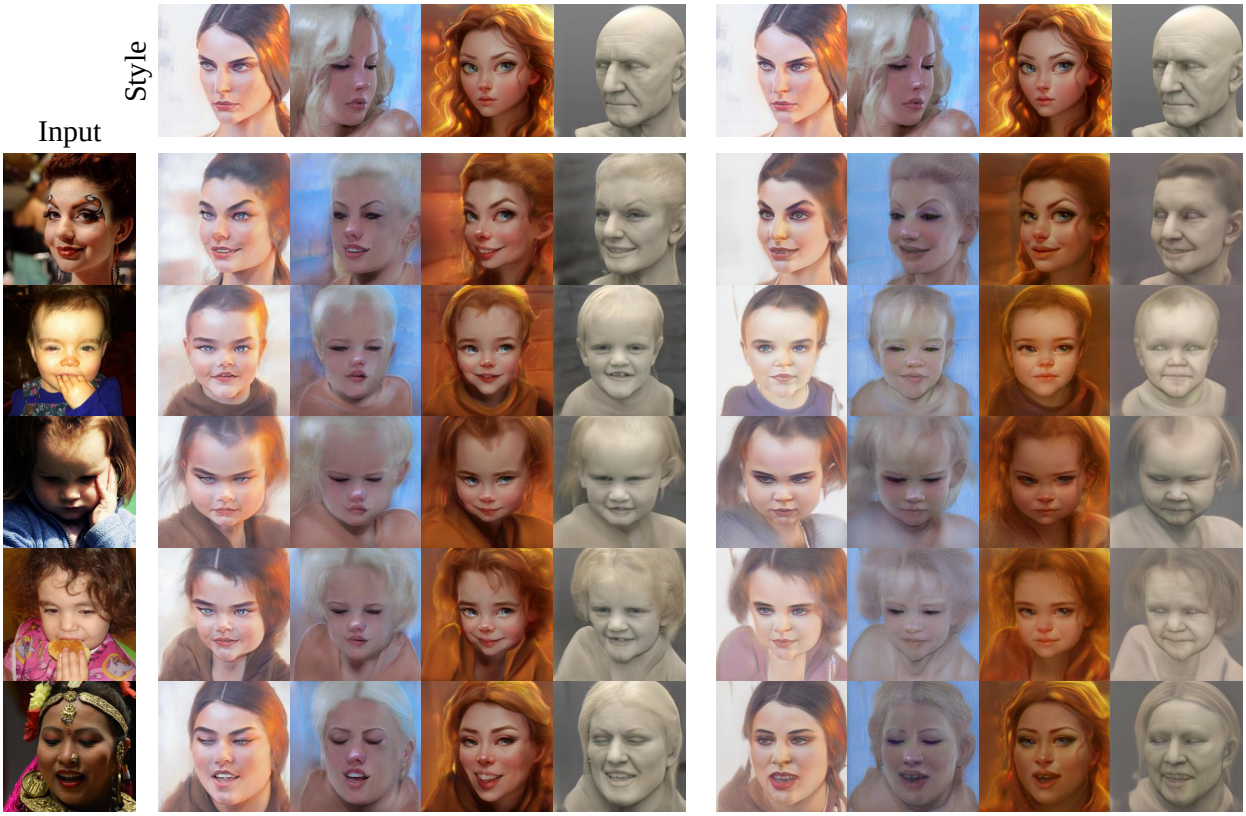

Figure 13. Stylization results of MTG and JoJoGAN.

	Methods	ArtFID↓			ID Similarity↑			Structure Similarity↓		
		AAHQ	MetFaces	Prev	AAHQ	MetFaces	Prev	AAHQ	MetFaces	Prev
High Density	MTG	34.54	**34.73**	35.18	0.2026	0.2562	0.2219	0.0403	0.0493	0.0477
	MTG+HFGI	35.03	37.19	36.23	0.3260	0.4362	0.3688	0.0357	0.0375	0.0370
	JoJoGAN	41.93	43.99	36.95	0.3783	0.3477	0.3280	0.0389	0.0465	0.0441
	JoJoGAN+HFGI	41.20	43.32	39.29	0.4927	0.4921	0.4353	0.0327	0.0385	0.0346
	DiffuseIT	44.03	52.92	46.56	0.6922	0.7259	0.6970	**0.0254**	**0.0252**	**0.0250**
	InST	**31.84**	46.13	30.69	0.1760	0.1864	0.1815	0.0390	0.0326	0.0383
	OSASIS(Ours)	33.06	41.66	**30.46**	**0.7191**	**0.7520**	**0.7303**	0.0367	0.0350	0.0345
Low Density	MTG	36.19	**36.52**	35.93	0.2228	0.2516	0.2263	0.0608	0.0557	0.0574
	MTG+HFGI	36.39	38.02	37.27	0.3730	0.4656	0.4063	0.0386	0.0350	0.0360
	JoJoGAN	43.51	45.23	38.49	0.3763	0.3579	0.3319	0.0589	0.00631	0.0605
	JoJoGAN+HFGI	40.41	44.74	41.09	0.5145	0.5207	0.4743	0.0411	0.0454	0.0403
	DiffuseIT	44.93	53.35	48.18	**0.6992**	0.7158	0.6994	**0.0309**	0.0300	**0.0310**
	InST	38.16	50.33	35.86	0.2253	0.2188	0.2238	0.0492	0.0443	0.0488
	OSASIS(Ours)	**34.89**	43.20	**33.20**	0.6825	**0.7323**	**0.7029**	0.0361	**0.0295**	0.0391

Table 4. Quantitative comparison.

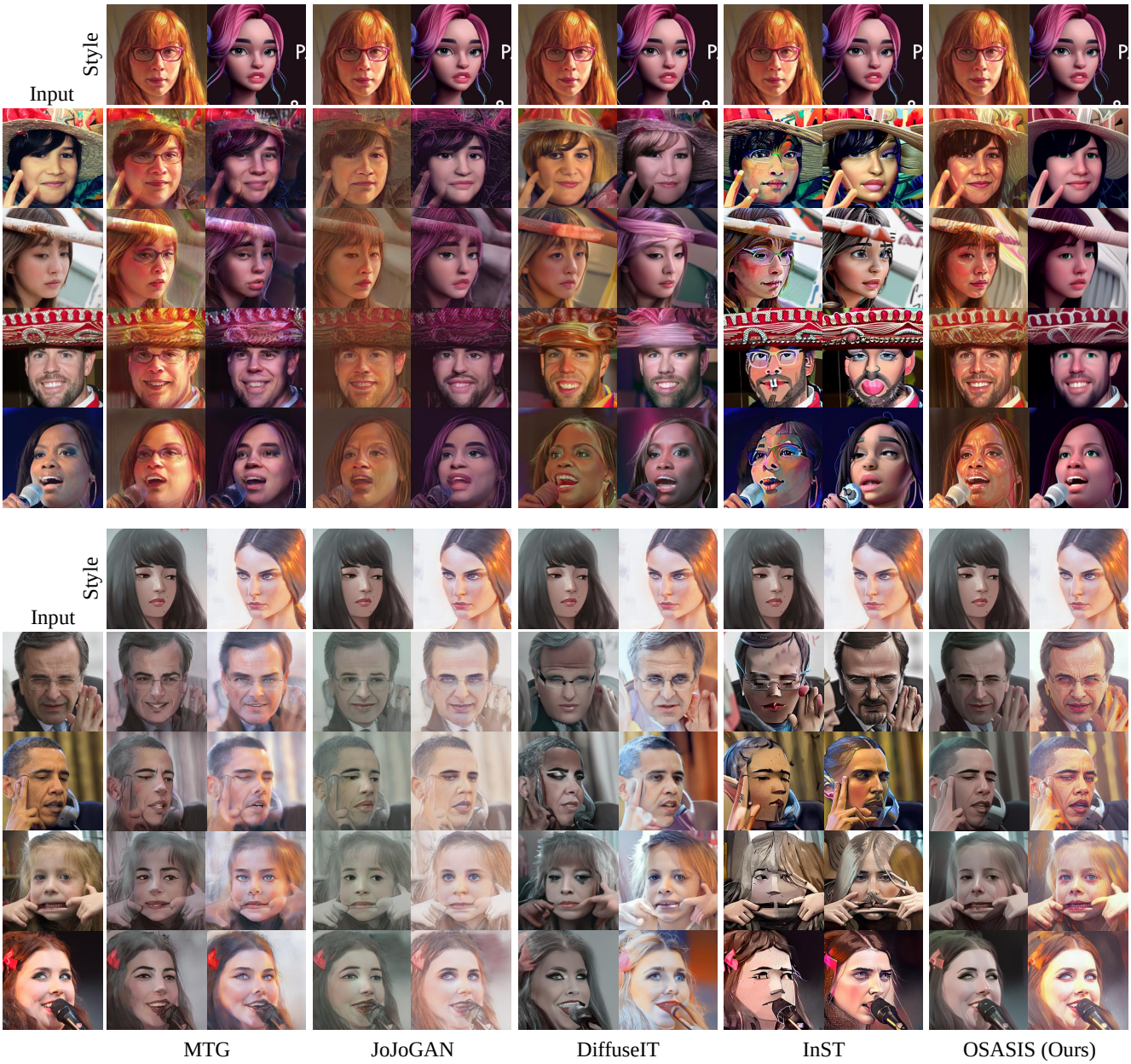

Figure 14. Stylization result of OSASIS and comparison methods.

Figure 15. Stylization result of OSASIS and comparison methods with OOD reference images.

References

[1] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6711–6720, 2021. 2

[2] Min Jin Chong and David Forsyth. Jojogan: One shot face stylization. In *European Conference on Computer Vision*, pages 128–152. Springer, 2022. 2

[3] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 2

[4] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2426–2435, 2022. 2

[5] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017. 2

[6] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4): 1–14, 2021. 2

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[8] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[9] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 2

[10] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 1

[11] 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 *International Conference on Learning Representations*, 2021. 2