Master: Meta Style Transformer for Controllable Zero-Shot and Few-Shot Artistic Style Transfer
–*Supplementary Material*–

Hao Tang[1*] Songhua Liu[2*] Tianwei Lin[3] Shaoli Huang[4]
Fu Li[3] Dongliang He[3] Xinchao Wang[2†]
[1]Center for Data Science, Peking University [2]National University of Singapore
[3]VIS, Baidu Inc. [4]Tencent AI Lab
tanghao@stu.pku.edu.cn songhua.liu@u.nus.edu
{lintianwei01,lifu,hedongliang01}@baidu.com shaolihuang@tencent.com xinchao@nus.edu.sg

Setting	Speed (sec. / image)	# of Param. (M)
StyTr2	0.087	25.14
Ours-L1	0.024	10.75
Ours-L3	0.030	10.75
Ours-L5	0.038	10.75

Table 1. Comparisons on inference speed and number of parameters at different settings. StyTr2 adopts 3 Transformer layers by default. For our method, L1/L3/L5 means using 1/3/5 Transformer layers in the test time.

k	1	2	3	4
$\mathcal{L}_{sty}$	3.389	2.661	2.384	1.811

Table 2. Impact of the number of inner optimization times k in the meta training on the style loss in the fast adaptation.

In this document, we provide more experimental analysis and results of the proposed meta style transformer (Master) for controllable zero-shot and few-shot artistic style transfer. We first compare our model with the existing Transformer-based methods in terms of efficiency. Then, we provide some qualitative analysis and ablation studies to the meta training and fast adaptation algorithms. Finally, we supplement more comparisons with more state-of-the-art techniques, more zero-shot and few-shot style transfer results, more examples of controlling the stylization via stacking different numbers of Transformer layers, more results of text-guided style transfer, and more extensions.

*Equal contribution.
†Corresponding author.

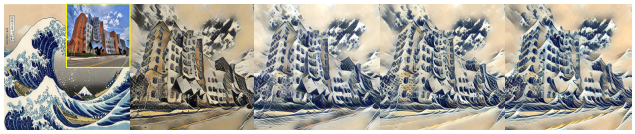


Figure 1. Impact of the number of fast adaptation iterations.

A. Efficiency

In this part, we compare the proposed Master model with the state-of-the-art Transformer-based style transfer method StyTr2 [3], in terms of inference speed and number of parameters. We experiment with stacking 1, 3, and 5 Transformer layers in the test time and comparisons at different settings are shown in Tab. 1. Here, StyTr2 adopts 3 Transformer layers by default and comparisons are conducted under 512×512 resolution. The speed is measured over 220 inference times and the same workstation with a Nvidia 3090 GPU is adopted as the platform for all settings.

Through the results, we can observe that the proposed model can have more than $2\times$ FPS compared with StyTr2, even when the number of Transformer layers is 5. Moreover, since parameters are shared across different Transformer layers, the total number of parameters would not increase with the increasing number of stacked layers and it is always significantly less than that of StyTr2. Thus, compared with existing Transformer-based models, Master achieves superior quality and efficiency simultaneously.

B. Meta Training and Fast Adaptation

Meta Training. Alg. 1 of the main paper shows the workflow of the meta training procedure, and the number of inner optimization times k is a hyper-parameter. As a

Figure 2. Few-shot stylization results under different base models: Our Master, StyFormer, and StyTr2.

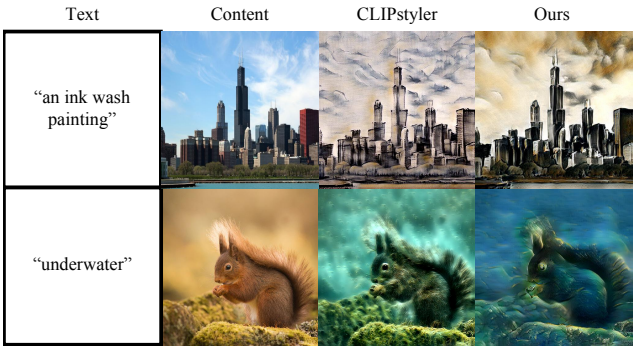


Figure 3. More comparisons on text-guided style transfer with Clipstyler.

meta learning algorithm, it requires k to be greater than 1. Otherwise, it would be degraded into pretraining and fine-tuning [11], which is essentially equivalent to the typical training pipeline in arbitrary style transfer and is the setting in our zero-shot style transfer. The larger the k is, the higher order of optimization procedures in few-shot learning can be learned, which may contribute to faster adaptation in the few-shot stage. Results of Tab. 2, which show the average style loss at the 10th iteration of fast adaptation, validate this effect.

Fast Adaptation. We provide example results of different numbers of fast adaptation iterations in Fig. 1. More local and global style patterns are captured by our model with the progress of fast adaptation, which suggests that our method potentially supports user-customized level of stylization by controlling the number of iterations during fast adaptation. Specifically, in all our experiments, we adopt 100 as the default number of iterations during the fast adaptation stage.

To further demonstrate the advantage of the Transformer model, we change our base model from our architecture to StyFormer and StyTr2 respectively and provide qualitative examples by these base models in Fig. 2, as a supplement to the training analysis in Fig. 7 of the main paper. We observe that our model renders global and local style patterns better.

C. More Results

C.1. Full Comparison Results

In order to further demonstrate the advantages of our proposed method, we provide more comparisons between our results with those by more state-of-the-art methods, as a supplement to Fig. 4 in the main paper. Here, there are 3 global transformation based methods (AdaIN [5], Linear style transfer [9], and MCCNet [1]), 1 patch swap based method (Avatar-Net [15]), 3 attention based methods (SANet [12], MANet [2], and AdaAttN [10]), 2 transformer based methods (StyTr2 [3] and StyFromer [17]), 2 meta learning based methods (MetaNet [14] and MetaStyle [18]), and the per-style-per-model method by Johnson *et al.* [6]. The comparisons are shown in Fig. 7 and the conclusions are consistent with those in the main paper:

- Global transformation based methods are not powerful enough to capture local style details.
- The patch based method Avatar-Net distorts major content structures heavily.
- Attention based methods are prone to either dirty textures, *e.g.*, SANet and MANet, or shallow style pattern migration, *e.g.*, AdaAttN.
- Following the design of vanilla Transformer, similar problems of dirty textures and content distortion also exist in StyTr2, *e.g.*, 4th, 5th, 6th, and 10th columns. Moreover, without leveraging local transformation, its performance on migration of local textures is not satisfactory enough, *e.g.*, 1st, 2nd, 3rd, 7th, 8th, 9th, and 11th columns.
- Compared with StyFormer, the local self-attention mechanism in our model extracts and transfers style patterns more sophisticatedly.
- It seems hard for MetaNet to be robustly adapted for a style image in a few shots.
- Results by MetaStyle often demonstrate shallower stylized effects compared with ours.
- Johnson *et al.* tends to fill content images with the learned style textures, which may also distort content structures. The similar effect also exists in the comparison results with the seminal optimization-based solution by Gatys *et al.* [4] as shown in Fig. 5(c) and Tab. 3.

Our method addresses above problems by dedicated self attention and cross-modality attention mechanisms with learnable and dynamic scaling parameters, which lead to more robust and vivid stylization results.

C.2. More Content-Style Pairs

To further illustrate the performance of our Master model, we provide more content-style pairs in Fig. 8. In each entry, upper and bottom images are results under zero-shot and few-shot settings respectively. Here, 1 Transformer layer is adopted. These results better demonstrate the robustness of our method to different kinds of content and style images.

C.3. More Controllable Style Transfer Results

We provide more controllable style transfer results by using different numbers of stacked Transformer layers in the inference time. As shown in Fig. 9, with more Transformer layers executed, the degree of stylization increases in general, where more intensive and vivid global and local style patterns are migrated. Quantitatively, we visualize the effect of tuning the number of Transformer layers in Fig. 4, which demonstrates that the trade-off between content loss and style loss can be controlled by this factor.

C.4. More Text-Guided Style Transfer Results

As a supplement to Fig. 6 in the main manuscript, in Fig. 3, we provide more qualitative comparison with Clipstyler [8], the state-of-the-art text-guided style transfer technique based on the per-text-per-model fashion. The conclusion is consistent with that in the main paper. We also visualize more pair-wise results of different texts and content images in Fig. 10.

C.5. More Ablation Results

Architecture: We provide more ablation results to better support the necessity of key designs in our Master model: using learnable scaling parameters for cross-attention, removing normalization in style encoder, and only updating style encoder in the few-shot training stage. The results are shown in Fig. 11, as a supplement to Fig. 6 in the main paper. Through the results, we can observe:

- Vanilla Transformer without learnable scaling parameters tend to distort original content structures. Such effects are obvious in background areas with less variation on textures.
- Using normalization in style encoder is harmful for stylization effects, since second-order statistics removed by the normalization contain important style information.
- Updating the whole model in the few-shot stage makes the training more difficult and leads to inferior stylization effects, compared with the case of only updating style encoder.

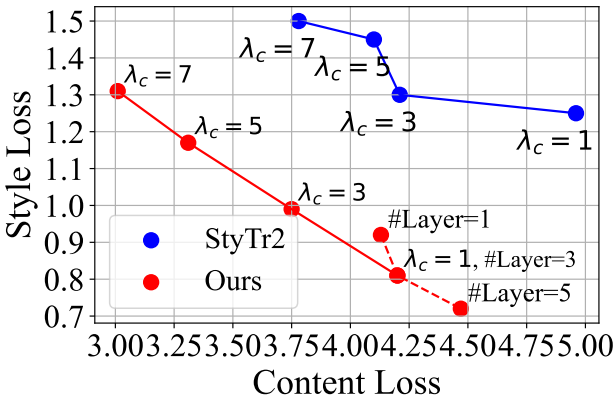
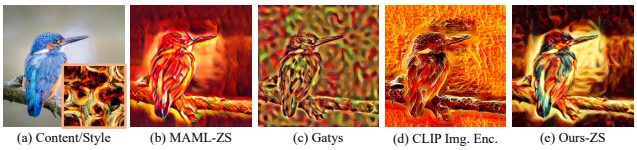


Figure 4. Quantitative comparisons with StyTr2 under different configurations of content weight.

Hyper-Parameter: For a fair comparison, we compare with StyTr2 [3], the vanilla Transformer model for style transfer, under the same configuration of loss function, *i.e.*, the same content weight, denoted as λ_c. The default λ_c in this paper is 1 while that in StyTr2 is 7, and the quantitative results are shown in Tab.1 of the main paper. The results under the same λ_c are provided in Fig. 4, where the superiority of our method can be reflected more clearly.

Training Algorithm: Compared with MetaStyle [18], a MAML-based few-shot style transfer method, our method has two major differences: the training algorithm is based on Reptile and the architecture is a novel Transformer model. We provide a fine-grained ablation study in Fig. 5 and Tab. 3, both qualitatively and quantitatively, to reflect the contribution of each component. In fact, both the model and the algorithm make improvement: the Transformer model mainly improves the stylization quality compared with existing models while Reptile mainly improves the training efficiency compared with MAML in MetaStyle. On the one hand, as shown in Fig. 8 of the main paper, replacing Master with vanilla Transformers would result in inferior quantitative metrics. On the other hand, we tried using MAML instead of Reptile before and found that it requires more time for convergence: 3 days for MAML v.s. 5 hours for Reptile. The computation of higher-order gradients increases the training difficulty, which further results in inferior performance as shown in Fig. 5(b) and Tab. 3. We also include ArtFID [16], a recently proposed metric for artistic style transfer, for better illustration.

Encoder: Our method adopts CLIP [13] to achieve text-guided style transfer, which contains an image encoder and text encoder. We use the image encoder for training and adopt the text encoder for inference, leveraging the aligned feature spaces of corresponding images and texts. In fact, it is also feasible to use the CLIP image encoder for image style transfer, rather than the Swin encoder by default. An



(a) Content/Style (b) MAML-ZS (c) Gatys (d) CLIP Img. Enc. (e) Ours-ZS

Figure 5. More qualitative ablation studies.

example is shown in Fig. 5(d). Since CLIP only returns a 512-d feature vector for an image, it mainly transfers the style globally and the performance on local details is inferior. Thus, Swin is used for image style transfer by default.

Content-Distortion Problem: We provide a more specific example to illustrate the content-distortion problem by the vanilla Transformer model. Assume that there are two 2-d content features: $c_1 = [0.5, 1]$ and $c_2 = [4, 1.5]$, two style features: $s_1 = [3.5, 0]$ and $s_2 = [-5, -5]$. Attention scores after Softmax are close to 1 for both c_1 and c_2 to s_1, and are close to 0 for both c_1 and c_2 to s_2. The transferred results with residual connection are $cs_1 = [4, 1]$ and $cs_2 = [7.5, 1.5]$, and the cosine similarity between c_1 and c_2 becomes 1 from 0.73. Thus, the original content-wise similarity is distorted. In this case, re-scaling content features by a factor larger than 1 may alleviate the drawback. This factor is made learnable in this paper and the model is provided with an opportunity to learn how to preserve the similarity in training and convergence. The metric $\mathcal{L}_{sim}$ in Eq. 9 quantifies this effect and experiments in Tab. 1 of the main paper demonstrate the effectiveness of our solution.

Impact of Multiple Transformer Layers on Training Convergence: One drawback of the vanilla Transformer model in style transfer is that the multi-layer structure can lead to difficult training convergence. As shown in Fig. 6, with more layers adopted, the loss may converge more slowly, and it even fails in the 5-layer case. There seems to be a contradiction with the conclusion on the generative model focusing on StyleGAN [7]: the model becomes more robust with more parameters. In fact, instead of generating new contents unconditionally in StyleGAN, style transfer aims to preserve contents and migrate style patterns at the same time. Stacking more layers in Transformer models may increase the complexity of the transfer function and tends to learn more abstract information. Thus, with more layers, it becomes harder to preserve original content structures during training. Sharing parameters for different layers kills three birds with one stone: it makes a light-weight, easy-to-train, and easy-to-control model.

C.6. More Extensions

Style Interpolation. Our model also supports style interpolation by conducting linear interpolation to a couple of output features of our Style Transformer. Two examples are shown in Fig. 12.

		$\mathcal{L}_{cont}\downarrow$	$\mathcal{L}_{sty}\downarrow$	ArtFID↓
Gatys *et al.*		4.24	1.67	37.24
StyTr2 (Same λ_c)		4.96	1.25	40.49
MAML	ZS	4.95	2.36	38.14
	FS	4.80	0.79	34.47
Ours	ZS	**4.20**	**0.81**	**32.80**
	FS	**4.24**	**0.79**	**32.70**

Table 3. More quantitative ablation studies.

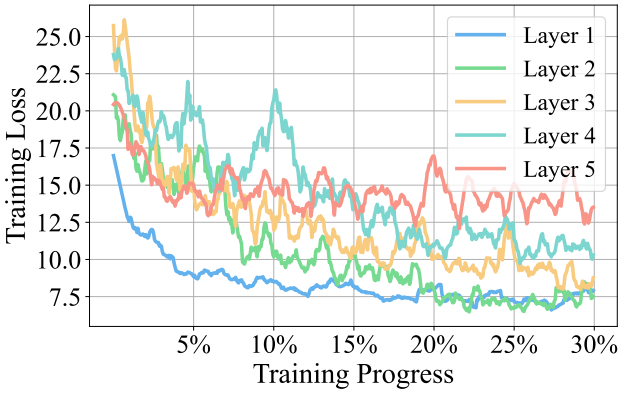


Figure 6. Fine-grained ablation studies on the number of layers used without parameter sharing to train a style transfer model.

Multi-Style Transfer. It is convenient for our method to achieve multi-style transfer by simply send features of multiple style images to the style encoder of our Master model. Results are shown in Fig. 13.

References

[1] Yingying Deng, Fan Tang, Weiming Dong, Haibin Huang, Chongyang Ma, and Changsheng Xu. Arbitrary video style transfer via multi-channel correlation. *arXiv preprint arXiv:2009.08003*, 2020. 2

[2] Yingying Deng, Fan Tang, Weiming Dong, Wen Sun, Feiyue Huang, and Changsheng Xu. Arbitrary style transfer via multi-adaptation network. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2719–2727, 2020. 2

[3] Yingying Deng, Fan Tang, Xingjia Pan, Weiming Dong, Chongyang Ma, and Changsheng Xu. $stytr^2$: Unbiased image style transfer with transformers, 2021. 1, 2, 3

[4] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[5] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017. 2

[6] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 2

[7] 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/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 4

[8] Gihyun Kwon and Jong Chul Ye. Clipstyler: Image style transfer with a single text condition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18062–18071, 2022. 3

[9] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast arbitrary style transfer. *arXiv preprint arXiv:1808.04537*, 2018. 2

[10] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Meiling Wang, Xin Li, Zhengxing Sun, Qian Li, and Errui Ding. Adaattn: Revisit attention mechanism in arbitrary neural style transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6649–6658, October 2021. 2

[11] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms, 2018. 2

[12] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5880–5888, 2019. 2

[13] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 3

[14] Falong Shen, Shuicheng Yan, and Gang Zeng. Meta networks for neural style transfer, 2017. 2

[15] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8242–8250, 2018. 2

[16] Matthias Wright and Björn Ommer. Artfid: Quantitative evaluation of neural style transfer. In *Pattern Recognition: 44th DAGM German Conference, DAGM GCPR 2022, Konstanz, Germany, September 27–30, 2022, Proceedings*, pages 560–576. Springer, 2022. 3

[17] Xiaolei Wu, Zhihao Hu, Lu Sheng, and Dong Xu. Styleformer: Real-time arbitrary style transfer via parametric style composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14618–14627, 2021. 2

[18] Chi Zhang, Yixin Zhu, and Song-Chun Zhu. Metastyle: Three-way trade-off among speed, flexibility, and quality in neural style transfer, 2019. 2, 3

Figure 7. Full comparison results as a supplement to Fig. 4 in the main paper. Zoom in for better details.

Figure 8. More content-style pairs. Upper and bottom images of each entry are results under zero-shot and few-shot settings respectively. Zoom in for better details.

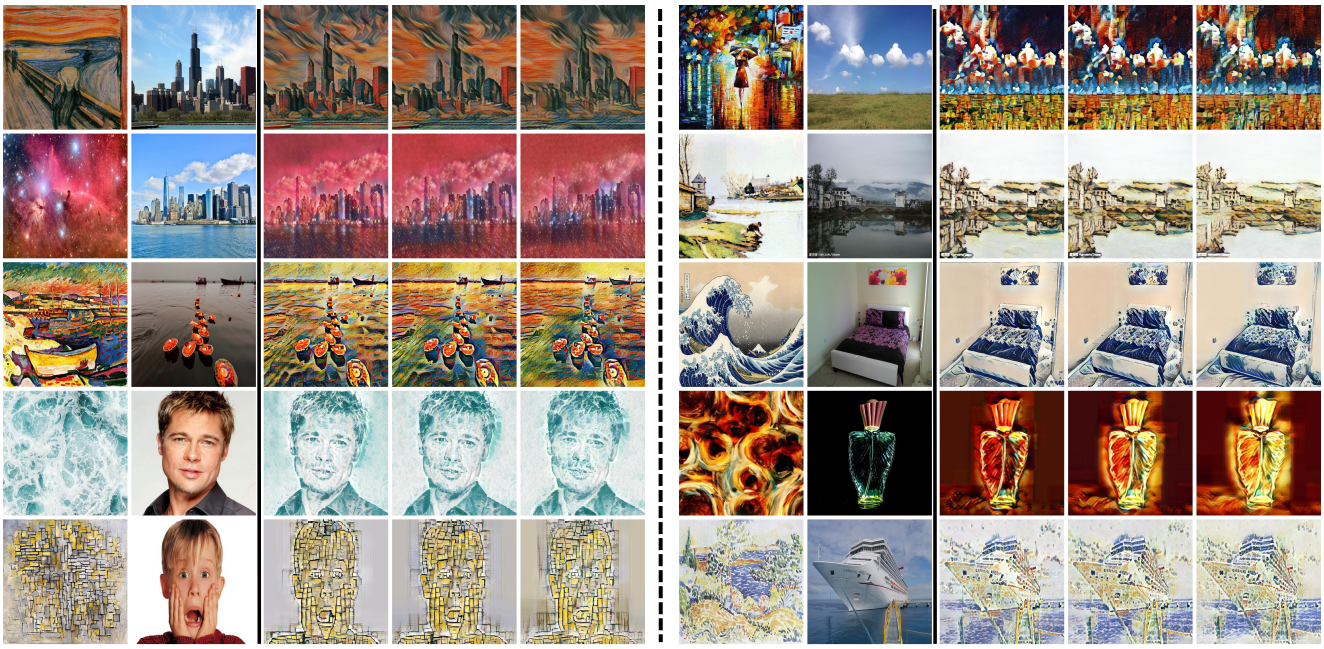


Figure 9. More controllable style transfer results by using different numbers of stacked Transformer layers in the test time.

Figure 10. More content-text pairs for text-guided style transfer.

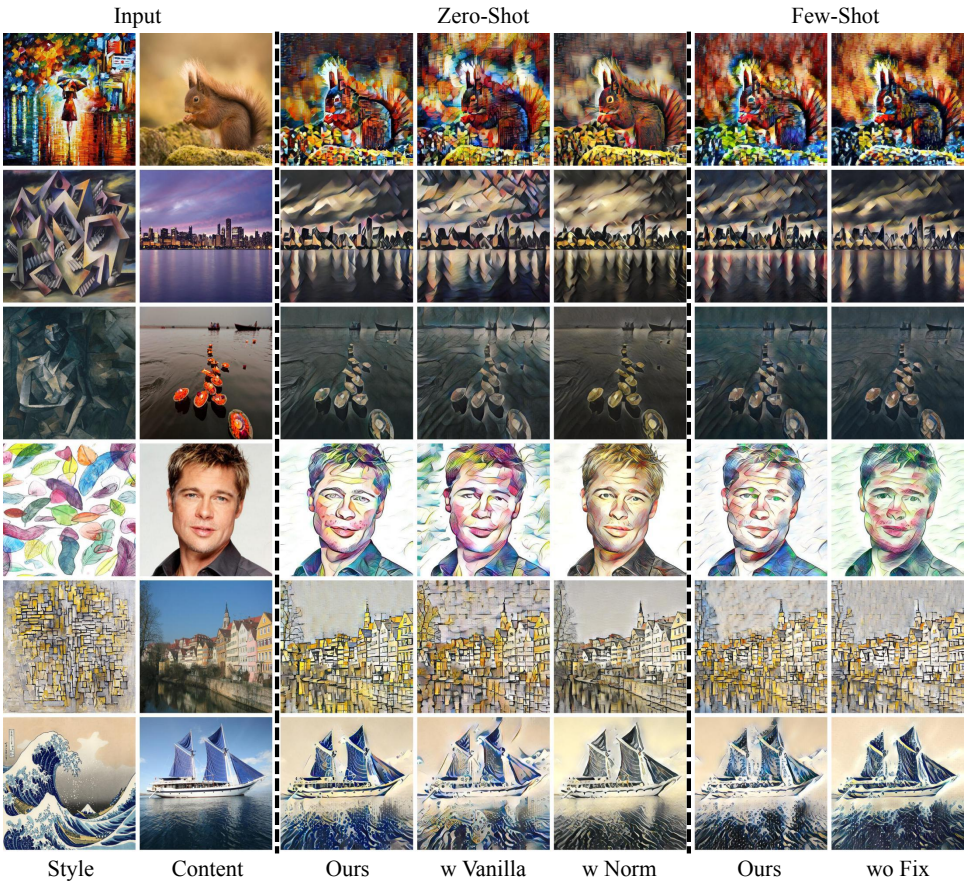


Figure 11. More ablation results as a supplement to Fig. 7 in the main paper. Zoom in for better details.

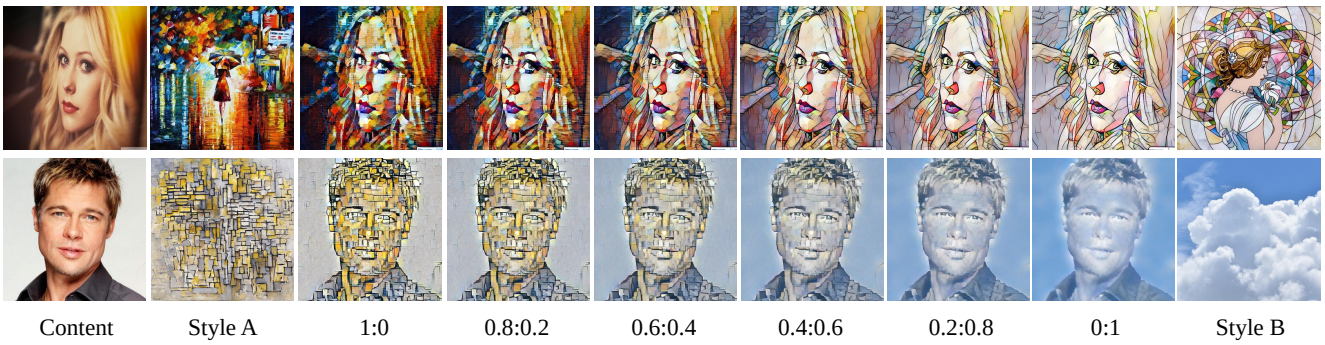


Figure 12. Two-style interpolation results. The content image and style images are shown on the two ends

Figure 13. Results of multi-style transfer.