Learning Graph Neural Networks for Image Style Transfer
– *Supplementary Material* –

Yongcheng Jing[1], Yining Mao[2], Yiding Yang[3], Yibing Zhan[4], Mingli Song[5,2], Xinchao Wang[3], and Dacheng Tao[1,4]

[1] The University of Sydney, Darlington, NSW 2008, Australia
[2] Zhejiang University, Hangzhou, ZJ 310027, China
[3] National University of Singapore, Singapore
[4] JD Explore Academy, China
[5] Zhejiang University City College, Hangzhou, ZJ 310015, China
xinchao@nus.edu.sg, dacheng.tao@gmail.com

We provide in this documents supporting materials that cannot fit into the manuscript due to page limit.

For our empirical validations, we include in this document:

- ***Two* newly-added ablation studies**, including the results with the proposed local patch-based manipulation (LPM) module and those without LPM. We also validate the effectiveness of the proposed GNN-based approach by developing two possible semi-parametric solutions and demonstrate here the corresponding comparative results;
- **Additional results of the *Five* ablation studies in the main paper**, including more results of heterogeneous neighborhood aggregation schemes, different distance metrics, various content and style patch sizes, distinct patch division schemes, and the designed intra-domain connections;
- **Additional results of the novel functionality**, including flexible diversified arbitrary stylization and multi-style amalgamation with a single model.

For our proposed approach, we provide here:

- **More architecture details** of each module;
- **More detailed explanations** of the proposed heterogeneous content-style and content-content message passing.

1 Architecture Details

We show in Tab. S1 the architecture details of the proposed method, corresponding to Fig. 2 in the main paper. In particular, for the image encoder, we use the first few layers of VGG-19 before `relu3_1`, as also done in [1], to generate more feature patches for matching. We do not include the global feature refinement module in Tab. S1, since there are no involved trainable parameters during our process of global feature refinement for the sake of computational efficiency. For the deformable module, we would like to clarify that we omit a sampling-interpolation procedure that is hard to depict in Tab. S1, which addresses the

fractional coordinate issue, with: 2304 = in_channels × sampling_window_size. Also, the outputs of the scale predictor in Tab. S1 are the final rescaled feature patches, with 6400 = in_chans ×patch_size^2.

Table S1. Detailed architectures of the image encoding module, deformable module, GNN-based local patch-based manipulation module, and feature decoding module in the proposed semi-parametric style transfer network, respectively, corresponding to Fig. 2 in the main paper.

	Layer	In_Chans	Out_Chans	Kernel	Stride	Activation
Image Encoding	Conv	3	3	1×1	1	-
	Conv	3	64	3×3	1	ReLU
	Conv	64	64	3×3	1	ReLU
	MaxPool	64	64	-	2	-
	Conv	64	128	3×3	1	ReLU
	Conv	128	128	3×3	1	ReLU
	MaxPool	128	128	-	2	-
	Conv	128	256	3×3	1	ReLU
Deformable Module	Conv	512	256	1×1	1	GELUS
	Conv	256	256	3×3	1	-
	Conv	256	6400	5×5	1	-
	Linear	6400	4	-	-	-
	Linear	2304	6400	-	-	-
Local Patch-based Manipulation	Feat2Patch	256	6400	-	-	-
	KNN	-	-	-	-	-
	GATConv	256	256	-	-	ReLU
	GATConv	256	256	-	-	ReLU
	Patch2Feat	6400	256	-	-	-
Feature Decoding	Conv	256	256	3×3	1	ReLU
	Conv	256	256	3×3	1	ReLU
	Conv	256	256	3×3	1	ReLU
	Conv	256	128	3×3	1	ReLU
	Upsample	128	128	1/2	-	-
	Conv	128	128	3×3	1	ReLU
	Conv	128	64	3×3	1	ReLU
	Upsample	64	64	1/2	-	-
	Conv	64	64	3×3	1	ReLU
	Conv	64	3	3×3	1	-

2 More Illustrations of Heterogeneous Style-Content and Content-Content Message Passing

In this section, we give more explanations of the proposed path-based message passing scheme that cannot fit into the main manuscript due to the page limit. We demonstrate the detailed style-to-content message passing and content-to-content message passing in Fig. S1, corresponding to Sect. 3.3 in the main manuscript. Specifically, the style-to-content message passing, as shown in Fig. S1, aims to aggregate style information from the k most similar style patches along the inter-domain edges (green arrows in Fig. S1). Subsequent to the style-to-content message passing, the proposed content-to-content aggregation further

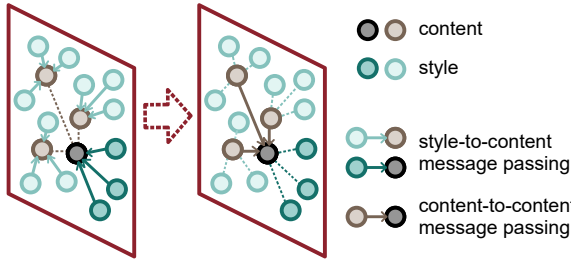

Fig. S1. Illustrations of the dedicated two-stage heterogeneous aggregation process, including style-to-content message passing stage (*i.e.*, the left red block in the figure) and content-to-content messing passing stage (*i.e.*, the right red block in the figure).

gathers the features from neighboring content nodes, such that the semantically-similar content regions will also be rendered with homogeneous style patterns. The effectiveness of such content-to-content message passing will be further validated in Fig. S6 of Sect. 4.3.

3 Newly-Added Ablation Studies

In this section, we perform extensive ablation studies to further validate the effectiveness of the proposed semi-parametric style transfer framework. In particular, we add ***two new ablation studies*** absent in the main manuscript due to the page limit, including the stylization results with the local patch-based manipulation (LPM) module and those without the LPM module, and also the

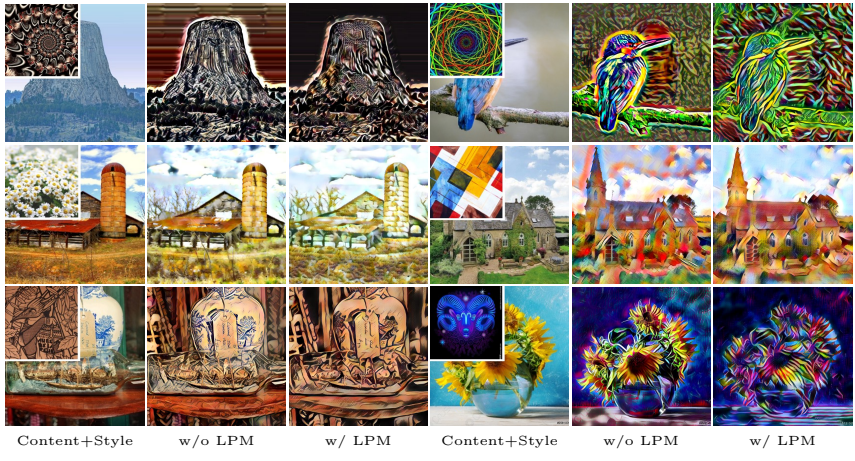

Fig. S2. Comparative results without the local patch-based manipulation (LPM) module and those with the LPM module.

results of the two possible solutions of semi-parametric stylization, including the combination of AdaIN and style swap, and also the combination of AdaIN and style decorator.

3.1 Stylization w/ and w/o Local Patch-based Manipulation Module

Fig. S2 shows the stylization results with the proposed local patch-based manipulation (LPM) module, and those without the LPM module. The stylized results without the proposed LPM module, as shown in the 2$^{\text{nd}}$ and the 5$^{\text{th}}$ columns of Fig. S2, retain the global appearance of the style images, but are prone to undesired local artifacts. In contrast, the results with the dedicated LPM are effective in producing fine-grained patterns and sharper details, as can be observed in the 3$^{\text{rd}}$ and the 6$^{\text{th}}$ columns Fig. S2. For example, the 3$^{\text{rd}}$ row, 6$^{\text{th}}$ column of Fig. S2 successfully transfers the corresponding style strokes to the petals, whereas the 5$^{\text{th}}$ column in Fig. S2 only keeps the original petal colors. Similar observations can also be obtained from the 1$^{\text{st}}$ row of Fig. S2, where the bird feathers are rendered with the corresponding best-matched style patterns.

3.2 Ours *vs* AdaIN+Style-Swap *vs* AdaIN+Style-Decorator

To further demonstrate the superiority of the proposed local patch-based manipulation module, we develop two possible solutions for semi-parametric neural

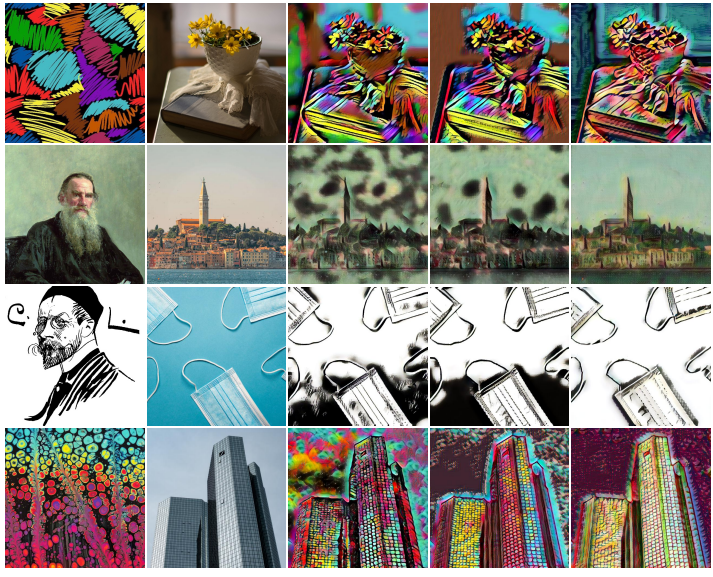

Fig. S3. Comparative results of the proposed GNN-based method with two possible semi-parametric solutions of AdaIN+Style-Swap and AdaIN+Style-Decorator.

style transfer. Specifically, we combine the style swap module in [1] with the global feature refinement module (*i.e.*, AdaIN), and also combine the style decorator module in [4] with AdaIN. As such, both local manipulation and global refinement are performed, leading to the two possible semi-parametric stylization methods. The results of the developed two possible solutions and our method (*i.e.*, AdaIN+GNN) are provided in Fig. S3, indicating that the proposed GNN-based approach is indeed superior than others.

4 Additional Results of Ablation Studies

In particular, we provide additional results of the ***five ablation studies*** that are introduced in the main manuscript, including the stylization results of various content/style patch sizes and heterogeneous aggregation mechanisms. We also give more results to validate the effectiveness of the proposed content-to-content message passing, the proposed deformable scheme, and the adopted similarity measurement metric of normalized cross-correlation. The results in this section correspond to Sect. 4.3 of the main paper.

4.1 Heterogeneous Aggregation Schemes: GAT *vs.* GCN *vs.* EdgeConv *vs.* GraphSage *vs.* GIN

We provide in Fig. S4 the results of using various GNN mechanisms in the proposed local patch-based manipulation module, including graph attention network

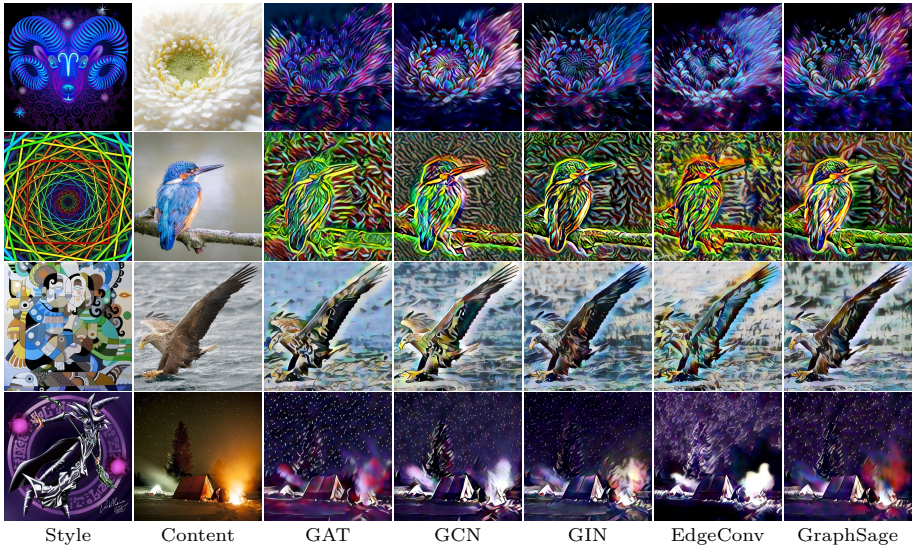

Fig. S4. Comparative results of using various aggregation mechanisms for heterogeneous message passing, including graph attention network (GAT) [5], graph convolutional network (GCN) [3], graph isomorphism network (GIN) [8], dynamic graph convolution (EdgeConv) [6], and GraphSage [2].

(GAT) [5], graph convolutional network (GCN) [3], dynamic graph convolution (EdgeConv) [6], GraphSage [2], and graph isomorphism network (GIN) [8]. In what follows, we start by briefly introducing these different GNN schemes and then explain the corresponding comparison results.

The simplest GNN mechanism is GCN, which iteratively optimizes node features via an isotropic averaging operation over the neighborhood node features: $h_i^{\ell+1} = \text{ReLU}\Big(U^\ell \ \text{Mean}_{j\in\mathcal{N}_i} \ h_j^\ell\Big)$, where $h_i^{\ell+1}$ represents the updated node features at layer $\ell+1$. U is the learnable transformation matrix. j denotes the neighbors $\mathcal{N}_i$ of the node i. GAT improves the vanilla by introducing the use of self-attention, as already introduced in Sect. 3.3 of the main manuscript. Also, GraphSage improves the simple GCN model by explicitly incorporating each node's own features from the previous layer, formulated as: $\hat{h}_i^{\ell+1} = \text{ReLU}\Big(U^\ell \ \text{Concat}\left(h_i^\ell, \ \text{Mean}_{j\in\mathcal{N}_i} \ h_j^\ell\right)\Big)$, where Concat denotes the concatenation operation. Moreover, the GIN architecture is based on the Weisfeiler-Lehman Isomorphism Test [7] to study the expressive power of GNNs, whereas EdgeConv generates the edge features that describe the relationships between a point and its neighbors for the subsequent information aggregation. More details can be found in [3, 5, 6, 2, 8].

As can be observed in Fig. S4, the GAT mechanism generally yield superior locally-style-aligned stylized results, thanks to its attention-based scheme. For example, in the 1$^{\text{st}}$ row of Fig. S4, GAT yields fine-grained style elements for the petals, in contrast to other GNNs that merely transfer the global style appearance from the target style. Another observation from Fig. S4 is that the GraphSage architecture is more effective at preserving the semantics of the content images, possibly due to its property of combining the node's own features from the previous layer. Also, the results of EdgeConv are less appealing, demonstrating that the edge features are inferior to the node features for the specific task of style transfer. Similar observations can be obtained from the results of GIN, where the style patterns are sometimes not sufficiently transferred to the stylized image, as shown in the 6$^{\text{th}}$ row of Fig. S4.

4.2 Distinct Patch Division Schemes

We show in Fig. S5 additional comparative results of equal-size patch division method, and those with the proposed deformable patch splitting scheme. Our deformable scheme allows for cross-scale style-content matching, thereby leading to spatially-adaptive multi-stroke stylization with the enhanced semantic saliency. Also, the proposed deformable module reduces the undesired artifacts in the stylization results, as shown in Fig. S5.

4.3 NST Graph w/ and w/o Intra-domain Edges

To validate the effectiveness of the proposed content-to-content message passing scheme, we perform extensive ablation studies in Fig. S6 by using or removing

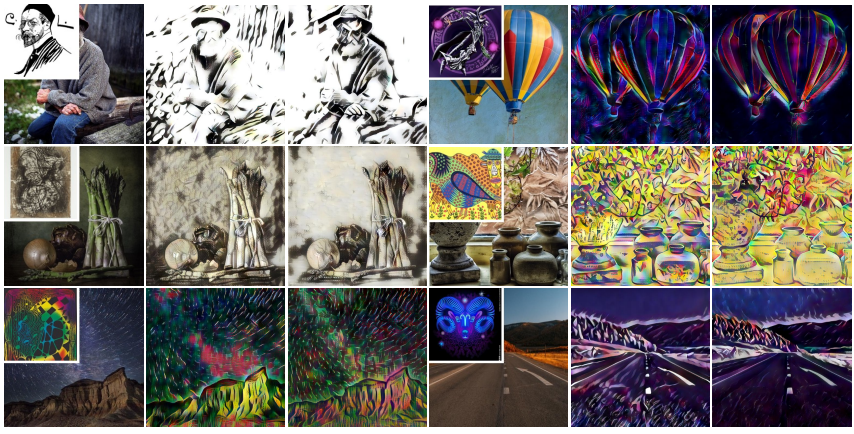

Fig. S5. Additional results of the equal-size patch division method and the proposed deformable module with a learnable scale predictor.

Fig. S6. Stylization results of removing the content-to-content intra-domain edges and those with the intra-domain edges.

the intra-domain edges in the constructed stylization graph. Our design of intra-domain edges, as shown in the 3$^{\text{rd}}$ and the 6$^{\text{th}}$ columns of Fig. S6, leads to more consistent style patterns in semantically-similar content regions, which is especially obvious when we observe the human faces in the 1$^{\text{st}}$ row of Fig. S6.

4.4 Euclidean Distance *vs.* Normalized Cross-correlation

In Fig. S7, we compare the results of using the Euclidean distance and the normalized cross-correlation (NCC) as the similarity measurement, respectively, in the construction of the stylization graph. The adopted metric of NCC in our framework, as observed from the 3$^{\text{rd}}$ and the 6$^{\text{th}}$ columns of Fig. S7, leads

Fig. S7. Results obtained using Euclidean distance and normalized cross-correlation (NCC) for similarity measurement during the construction of heterogeneous content-to-style and content-to-content edges.

to superior performance than the Euclidean distance (Fig. S7, the 2nd and 5th columns) in terms of both the global stroke arrangements and local details. We take the 3rd row of Fig. S7 as an example. It is evident that the stylization results with the Euclidean distance have more artifacts than those with NCC in

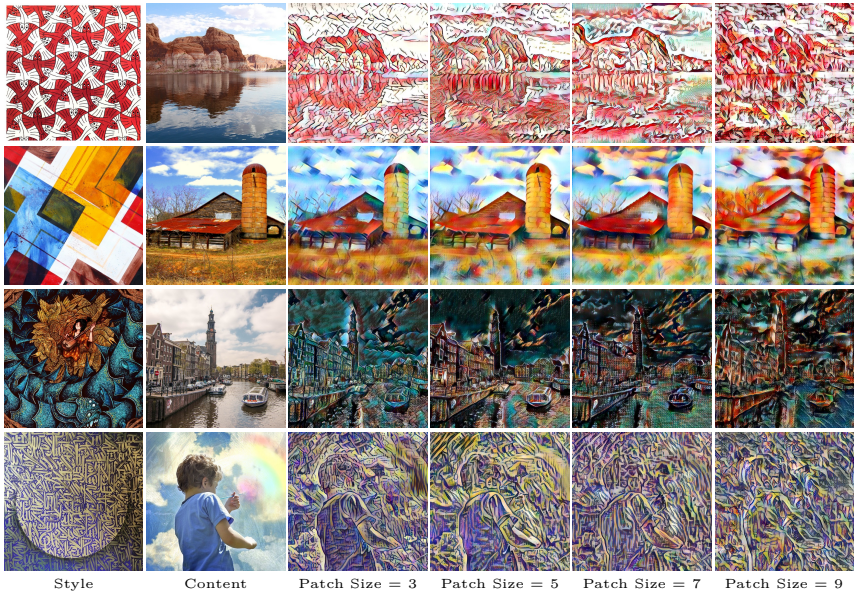

Fig. S8. Results obtained using various patch sizes for constructing content and style vertices in local patch-based manipulation module.

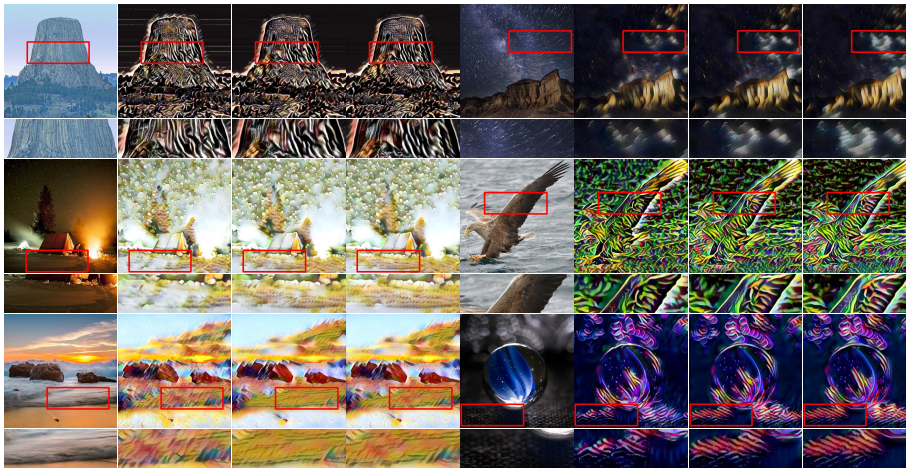

Fig. S9. Additional results of diversified patch-based arbitrary style transfer with solely a single model, corresponding to Fig. 6 in the main manuscript. We zoom in on the same regions (*i.e.*, the red frames) to observe the details.

the background of the sewing machine, demonstrating that NCC is better-suited for patch-based matching in our GNN-based framework.

4.5 Various Patch Sizes

We demonstrate in Fig. S8 the results of diversified feature patch sizes. Larger patch sizes, as shown in Fig. S8, generally lead to larger strokes in the stylized results. For example, the stylized images in the 3$^{\text{rd}}$ row, the 6$^{\text{th}}$ column of Fig. S8 has much larger strokes than those in the 3$^{\text{rd}}$ row, the 3$^{\text{rd}}$ column, which is especially obvious from the regions of the sky.

5 Additional Results of User Controls

5.1 Diversified Stylization Control

We provide in this section more results of the proposed diversified stylization control, corresponding to Fig. 9 in the main manuscript. Additional diversified results are given in Fig. S9, where we zoom in on the same regions in the red frames for the illustrations of local details. For example, in the last row of Fig. S9, it is noticeable that our diversified stylization control can yield various style patterns with different colors and strokes with only a single trained model. Such diversified user control is simply achieved by using different numbers of style-to-content connections during style-to-content message passing, leading to a limited auxiliary computational burden.

5.2 Multi-style Amalgamation

The proposed GNN-based style transfer approach also triggers the functionality of flexible multi-style transfer that combines the style patterns in multiple distinct artistic styles. We show in Fig. S10 the results that amalgamate four different style images as an example, but we would like to clarify that our method readily supports arbitrary style numbers for compositions. From the algorithm level, this multi-style image stylization is specifically realized by exploiting the style feature patches from multiple style images to construct the style vertices, which are then used to establish the *multistyle*-to-content heterogeneous connections for the subsequent *multistyle* message passing.

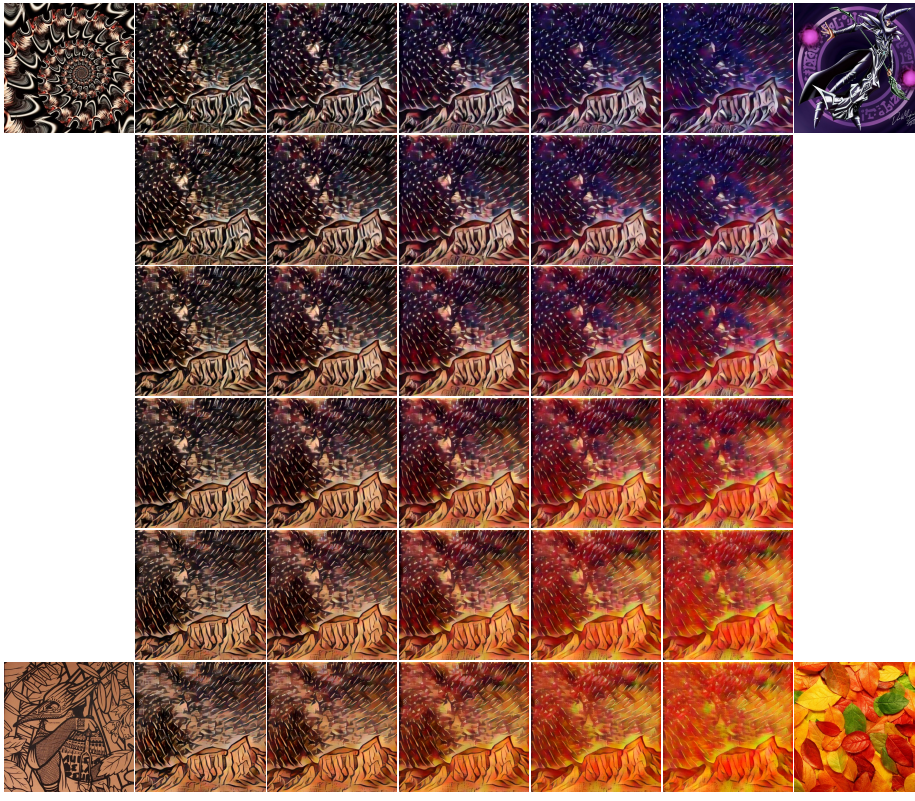

Fig. S10. Multi-style transfer within a single image, by performing style interpolation among various artistic styles.

References

1. Chen, T.Q., Schmidt, M.: Fast patch-based style transfer of arbitrary style. In: NeurIPS Workshop on Constructive Machine Learning (2016)

2. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: NeurIPS (2017)
3. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
4. Sheng, L., Shao, J., Lin, Z., Warfield, S., Wang, X.: Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In: CVPR (2018)
5. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. In: ICLR (2018)
6. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. TOG (2019)
7. Weisfeiler, B., Lehman, A.A.: A reduction of a graph to a canonical form and an algebra arising during this reduction. Nauchno-Technicheskaya Informatsia (1968)
8. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: ICLR (2019)