Learning Graph Neural Networks for Image Style Transfer

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

Abstract. State-of-the-art parametric and non-parametric style transfer approaches are prone to either distorted local style patterns due to global statistics alignment, or unpleasing artifacts resulting from patch mismatching. In this paper, we study a novel semi-parametric neural style transfer framework that alleviates the deficiency of both parametric and non-parametric stylization. The core idea of our approach is to establish accurate and fine-grained content-style correspondences using graph neural networks (GNNs). To this end, we develop an elaborated GNN model with content and style local patches as the graph vertices. The style transfer procedure is then modeled as the attention-based heterogeneous message passing between the style and content nodes in a learnable manner, leading to adaptive many-to-one style-content correlations at the local patch level. In addition, an elaborated deformable graph convolutional operation is introduced for cross-scale style-content matching. Experimental results demonstrate that the proposed semi-parametric image stylization approach yields encouraging results on the challenging style patterns, preserving both global appearance and exquisite details. Furthermore, by controlling the number of edges at the inference stage, the proposed method also triggers novel functionalities like diversified patch-based stylization with a single model.

Keywords: Neural style transfer · Graph neural networks · Attention-based message passing

1 Introduction

Image style transfer aims to automatically transfer the artistic style from a source style image to a given content one, and has been studied for a long time in the computer vision community. Conventionally, image style transfer is generally cast as the problem of non-photorealistic rendering in the domain of computer graphics. Inspired by the success of deep learning [9, 41, 8, 55, 10], Gatys *et al.* [11] pioneer the paradigm that leverages the feature activations

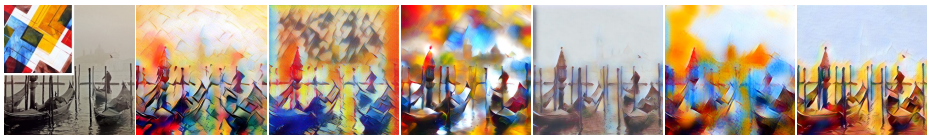

Fig. 1. Existing parametric [14, 1, 30] and non-parametric [6, 42] NST methods either barely transfer the global style appearance to the target [6], or produce distorted local style patterns [14, 1, 30] and undesired artifacts [42]. By contrast, the proposed GNN-based semi-parametric approach achieves superior stylization performance in the transfers of both global stroke arrangement and local fine-grained patterns.

from deep *convolutional neural networks (CNNs)* to extract and match the target content and style, leading to the benefits of no explicit restrictions on style types and no requirements of ground-truth training data. As such, various CNN-based style transfer methods are developed in the literature [22, 25, 5, 49, 47, 15, 13, 35, 34], establishing a novel field of *neural style transfer (NST)* [18].

State-of-the-art NST algorithms can be categorized into two streams of methods, parametric and non-parametric ones, depending on the style representation mechanisms. In particular, parametric NST approaches rely on the global summary statistics over the entire feature map from pre-trained deep CNNs to extract and match the target artistic style [11, 21, 14]. Non-parametric neural methods, also known as patch-based NST methods [6, 42], leverage the local feature patches to represent the style information, inspired by the conventional patch-based texture modeling approaches with Markov random fields. The idea is to swap the content neural patches with the most similar style ones, through a greedy one-to-one patch matching strategy.

Both parametric and non-parametric methods, unfortunately, have their own limitations, as demonstrated in Fig. 1. Parametric stylization methods achieve good performance in transferring the overall appearance of the style images, but are incompetent in generating fine-grained local style patterns. By contrast, non-parametric style transfer algorithms allow for locally-aligned stylization; however, such patch-based methods are typically accomplished with the undesired artifacts due to content-style mismatching.

In this paper, we present a semi-parametric style transfer scheme, towards alleviating the dilemmas of existing parametric and non-parametric methods. On the one hand, our semi-parametric approach allows for the establishment of more accurate many-to-one correspondences between different content and style regions in a learnable manner. As such, our approach explicitly tackles the issue of content-style mismatching in non-parametric NST algorithms, thereby largely alleviating the deficiency of unplausible artifacts. On the other hand, the proposed semi-parametric method adaptively divides content and style features into tiny and cross-scale feature patches for stylization, thus addressing the dilemma of lacking local details in prior parametric schemes.

Towards this end, we introduce to the proposed semi-parametric NST a dedicated learning mechanism, *graph neural networks* (GNNs), to enable adaptive local patch-level interplay between the content and style. As a well-established learning paradigm for handling non-Euclidean data, GNNs are designed to explicitly account for structural relations and interdependency between nodes. Moreover, GNNs are equipped with efficacious strategies for aggregating information from multiple neighbors to a center node. Such competences make GNN an ideal tool for tackling the intricate content-style region matching challenge in style transfer, especially the many-to-one mapping between each content patch and multiple potentially-matching style patches. We therefore exploit GNNs to adaptively set up the faithful topological correspondences among the very different content and style, such that every content region is rendered with the optimal style strokes.

Specifically, we start by building a heterogeneous NST graph, with content and style feature patches as the vertices. The multi-patch parametric aggregation in semi-parametric NST can thereby be modeled as the message passing procedure among different patch nodes in the constructed stylization graph. By employing the prevalent GNN mechanisms such as the graph attention network, the k most similar patches can be aggregated in an attention-based parametric manner. The aggregated patches are then composed back into the image features, which are further aligned with the target global statistics to obtain the final stylized result. Also, a deformable graph convolutional operation is devised, making it possible for cross-scale style-content matching with spatially-varying stroke sizes in a single stylized image. Furthermore, our GNN-based NST can readily perform diversified patch-based stylization, by simply changing the number of connections during inference.

In sum, our contribution is a novel semi-parametric arbitrary stylization scheme that allows for the effective generation of both the global and local style patterns, backed by a dedicated deformable graph convolutional design. This is specifically achieved through modeling the NST process as the message passing between content and style under the framework of GNNs. Experimental results demonstrate that the proposed GNN-based stylization method yields results superior to the state of the art.

2 Related Work

Neural Style Transfer. Driven by the power of *convolutional neural networks (CNNs)* [58, 62, 57, 26], Gatys *et al.* propose to leverage CNNs to capture and recombine the content of a given photo and the style of an artwork [11], leading to the area of *neural style transfer (NST)*. Existing NST approaches can be broadly divided into parametric and non-parametric NST methods. Specifically, parametric NST approaches leverage the global representations to transfer the target artistic style, which are obtained by computing the summary statistics in either an image-optimization-based online manner [11, 28, 40, 36], or model-optimization-based offline manner [21, 59, 29, 45, 1, 35, 3, 30, 14, 4, 16]. On

the other hand, non-parametric methods exploit the local feature patches to represent the image style [27, 42, 2, 6, 37, 31], inspired by the conventional patch-based texture modeling approaches with Markov random fields. The idea is to search the most similar neural patches from the style image that match the semantic local structure of the content one [27, 42, 2, 6, 37, 31]. This work aims to seek a balance between parametric and non-parametric NST methods by incorporating the use of GNNs.

Graph Neural Networks. GNNs have merged as a powerful tool to handle graph data in the non-Euclidean domain [24, 43, 19, 20, 51, 54, 52, 53, 33]. In particular, the seminal work of Kipf and Welling [24] proposes graph convolutional networks (GCNs), which successfully generalizes CNNs to deal with graph-structured data, by utilizing neighborhood aggregation functions to recursively capture high-level features from both the node and its neighbors. The research on GNNs leads to increasing interest in deploying GNN models in various graph-based tasks, where the input data can be naturally represented as graphs [63]. Moreover, the emerging transformers can also be treated as generalizations of GNNs [39, 56, 61, 60, 50]. Unlike these existing works where the inputs are themselves non-grid graphs, we aim to extend the use of GNN models to effectively manipulate grid-structured images, such that various image-based tasks can be benefited from GNNs.

3 Proposed Method

Towards addressing the limitations of existing parametric and non-parametric NST methods, we introduce the proposed semi-parametric style transfer framework with GNNs. In what follows, we begin by providing an overview of the proposed GNN-based approach, and then elaborating several key components, including the construction of the topological NST graph, the dedicated deformable graph convolutional operation customized for the established NST graph, and the detailed 2-hop heterogeneous message passing process for stylization. Finally, we illustrate the cascaded patch-to-image training pipeline, tailored for the proposed GNN-based stylization system.

3.1 Network Overview

The overall workflow of the proposed semi-parametric NST framework is shown in Fig. 2. There are primarily four modules in the whole pipeline, termed as *image encoding*, *local patch-based manipulation*, *global feature refinement*, and *feature decoding*. At the heart of the proposed framework is the *local patch-based manipulation* module, which will be further detailed in the following sections.

Image Encoding Module. The proposed semi-parametric stylization starts by receiving style and content images as inputs and encoding these images into meaningful feature maps (the green and yellow blocks in Fig. 2), by exploiting the first few layers of the pre-trained VGG network. In particular, unlike the existing work [14] that uses the layers before `relu4_1`, we leverage the VGG

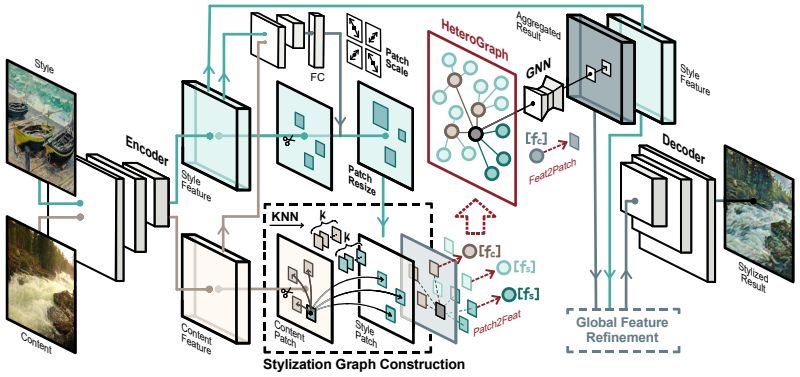

Fig. 2. Network architecture of the proposed semi-parametric style transfer network with GNNs. From left to right, the corresponding stylization pipeline comprises four subprocesses, *i.e.*, image encoding with the encoder, local patch-based manipulation based on heterogeneous GNNs, global feature refinement, and the feature decoding procedure. The symbols of scissors represent the process to divide the feature maps into feature patches. HeteroGraph denotes the established heterogeneous stylization graph with two types of content-style inter-domain connections and content-content intra-domain connections.

layers up to `relu3_1`, for the sake of more valid feature patches that can be exploited by the following local patch-based feature transformation stage.

Local Patch-based Manipulation Module. With the embeded content and style features as inputs, the local patch-based manipulation module extracts the corresponding content and style feature patches with the stride of s and the sliding window size of $p \times p$, represented as the scissor symbol in Fig. 2. We then build a heterogeneous stylization graph (the red frame in Fig. 2) with the obtained feature patches as graph nodes and perform the dedicated deformable graph convolution to generate the locally-stylized features, which will be further detailed in the succeeding Sect. 3.2 and Sect. 3.3.

Global Feature Refinement Module. The produced style-transferred results from the stage of patch-based manipulation are effective at preserving fine-grained local style patterns; however, the global style appearance is likely to be less similar to the target style image, due to the lack of global constraint on the stroke arrangement. To alleviate this dilemma, we propose a hierarchical patch-to-image stylization scheme to yield both the exquisite brushstroke and large-scale texture patterns. This is achieved by refining the feature representations at a global level, subsequent to the local patch-based manipulation. For the specific refinement method, since there already exist several effective global feature decorated strategies in the field of NST (*e.g.*, adaptive instance normalization (AdaIN) [14] and zero-phase component analysis (ZCA) [30]), here we directly utilize AdaIN as our refinement scheme, considering its high efficiency.

Feature Decoding Module. The last stage of our semi-parametric style transfer pipeline, termed as feature decoding, aims to decode the obtained feature representations from the preceding global feature refinement module into the final stylized image. The decoder module specifically comprises a sequence of convolutional and bilinear upsampling layers with the ReLU nonlinearities.

In the following sections, we will explain more details regarding the key module of *Local Patch-based Manipulation* with GNNs, including the graph construction procedure and the deformable graph convolutional process.

3.2 Stylization Graph Construction

At the stage of local patch-based manipulation, the first challenge towards the adaptive patch-level interactions between content and style with GNNs is the establishment of topological graphs. Unlike conventional GNN-based applications where the inputs can be naturally modeled as graphs (*e.g.*, biological molecules and social networks), there is no such natural topological structure for our task of semi-parametric image style transfer. To address this issue, we develop a dedicated graph construction technique, tailored for image stylization.

We start by giving the mathematical model of general graph-structured data as: $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{G}$ represents a directed or undirected graph. $\mathcal{V}$ denotes the set of vertices with nodes $v_i \in \mathcal{V}$. $\mathcal{E}$ represents the edge set with $(v_i, v_j) \in \mathcal{E}$, where $\{v_j\}$ is the set of neighboring nodes of v_i. Each vertex has an associated node feature $\mathcal{X} = [x_1\ x_2\ ...\ x_n]$. For example, x can be defined as the 3D coordinates in the task of point cloud classification.

As can be observed from the above formulation of prevalent graph data, the key elements in a graph are the vertices with the corresponding node features as well as the edges, which are thereby identified as our target objects to instantiate in the domain of style transfer as follows:

Heterogeneous Patch Vertices. To leverage GNNs to benefit the local-level stylization, we model in our framework the content and style patches as the graph nodes. Specifically, we exploit the content and style feature activations from the pre-trained VGG encoder, shown as the green and yellow blocks in Fig. 2, respectively, to capture the corresponding feature patches with a sliding window (*i.e.*, the scissor symbol in Fig. 2), in a similar manner as what is done when performing convolutions. We set the stride as 1 by default, meaning that there exist overlaps among our extracted activation patches. Such a manner of overlapped patch generation allows for smooth transitions among different stylized regions. In particular, to achieve cross-scale patch matching, we perform multi-scale patch division, which will be demonstrated in detail as a part of the deformable convolution in Sect. 3.3.

For the definition of the associated features for each patch vertex, we use a `Patch2Feat` operation, depicted as the red fonts in Fig. 2, to produce the desired format of node features for the use of the subsequent GNN layers, as also done in [64]. The designed `Patch2Feat` operation specifically amalgamates the c-dimensional features at each position of the $p \times p$ activation patch into a 1-dimensional feature vector, which is then considered as the node feature at

every patch vertex. The derived content and style node features are shown as $[f_c]$ and $[f_s]$ in Fig. 2, respectively, for the use of the latter GNN layers.

Inter- and Intra-KNN Edges. Another critical issue in building the stylization graph is the establishment of connections among different patch vertices. Customized for the task of style transfer, we formulate two types of edges, termed as *content-style inter-domain edges* and *content-content intra-domain edges*, leading to a special kind of heterogeneous graph.

In particular, the inter-domain connections between heterogeneous style and content nodes aim to attain more accurate many-to-one style-content matching for patch-based stylization. More specifically, for each content query patch $\phi_i(\mathcal{F}_c)$ with $\mathcal{F}_c$ representing the whole content feature map, we search the corresponding k-nearest ones in the set of style feature patches $\{\phi(\mathcal{F}_s)\}$, which are identified as the neighbors coupled with inter-domain edges. This process of k-nearest neighbor search (KNN) is shown in the black dotted frame in Fig. 2. We employ the distance metric of normalized cross-correlation (NCC) for pair-wise KNN, by scoring the cosine distance between a couple of content and style patches. Given a specific content patch $\phi_i(\mathcal{F}_c)$ as the query, our KNN procedure based on NCC can be specifically formulated as:

$$\mathrm{KNN}(\phi_i(\mathcal{F}_c), \{\phi(\mathcal{F}_s)\}) = \underset{j\in\{1,\ldots,N_s\}}{\arg\max_{\mathrm{k}}} \frac{\langle \phi_i(\mathcal{F}_c), \phi_j(\mathcal{F}_s)\rangle}{\|\phi_i(\mathcal{F}_c)\|\|\phi_j(\mathcal{F}_s)\|}, i \in \{1,\ldots,N_c\}, \quad (1)$$

where N_c and N_s denote the cardinalities of the corresponding content and style patch sets, respectively. $\max_{\mathrm{k}}$ returns the k largest elements from the set of the computed pair-wise NCCs. $\mathrm{KNN}(\phi_i(\mathcal{F}_c))$ represents the target k nearest-neighboring style vertices for the content patch $\phi_i(\mathcal{F}_c)$.

We also introduce the intra-domain connections within the set of content activation patches in our stylization graph, shown as the brown arrows in the black dotted rectangle in Fig. 2. The goal of such content-to-content edges is to unify the transferred styles across different content patches. In other words, we utilize the devised intra-domain connections to make sure that the semantically-similar content regions will also be rendered with homogeneous style patterns. This is specifically accomplished by linking the query content patch $\phi_i(\mathcal{F}_c)$ with the top-k most similar patches $\{\phi_j(\mathcal{F}_c)\}$ where $j \in \{1,\ldots,N_c\}$, by NCC-based KNN search in a similar manner with that in Eq. 1.

The ultimate heterogeneous stylization graph, with the two node types of content and style vertices and also the two edge types of inter- and intra-domain connections, is demonstrated as the red rectangle in Fig. 2. The relationship between the involved nodes is defined as the NCC-based patch similarity.

3.3 Deformable Graph Convolution

With the constructed stylization graph, we are then ready to apply GNN layers to perform heterogeneous message passing along the content-style inter-domain edges and also content-content intra-domain edges. A naïve way will be simply performing existing graph convolutions on the heterogeneous stylization graph to aggregate messages from the content and style vertices.

However, this vanilla approach is not optimal for the task of style transfer, due to a lack of considerations in feature scales. Specifically, in the process of image stylization, the proper feature scale is directly correlated with the stroke scale in the eventual output [17], which is a vital geometric primitive to characterize an artwork. The objective stylized results should have various scales of style strokes across the whole image, depending on the semantics of different content regions.

Towards this end, we propose a dedicated deformable graph convolutional network that explicitly accounts for the scale information in message passing. The devised deformable graph convolutional network comprises two components. Specifically, the first component is an elaborated *deformable scale prediction module*, with a fully-connected (FC) layer in the end, that aims to generate the optimal scale of each patch in a learnable manner before conducting message aggregation, as also done in [7]. In particular, the scale predictor receives both the content and style features as inputs, considering the potential scale mismatching between the content and style, as shown in the upper left part of Fig. 2.

As such, by adaptively performing scale adjustment according to both content and style inputs, the proposed deformable graph conventional network makes it possible for cross-scale style-content matching with spatially-varying stroke sizes across the whole image. We clarify that we only incorporate one-single predictor in our deformable graph convolutional network that produces the style scales, for the sake of computational efficiency. There is no need to also augment another predictor for content scale prediction, which is, in fact, equivalent to fixing the content scale and only changing the style one.

The second component of the proposed deformable graph convolutional network is the *general feature aggregation module* that learns to aggregate the useful features from the neighboring heterogeneous content and style nodes. Various existing message passing mechanisms can, in fact, readily be applied at this stage for message propagation. Here, we leverage the graph attention scheme to demonstrate the message flow along with the two types of stylization edges, which empirically leads to superior stylization performance thanks to its property of anisotropy.

Specifically, given an established stylization graph, our dedicated heterogeneous aggregation process is composed of two key stages, termed as *style-to-content message passing stage* and *content-to-content messing passing stage*:

Style-to-Content Message Passing. The first style-to-content stage aims to gather the useful style features from the k neighboring style vertices. For the specific message gathering method, one vanilla way is to treat the information from every style vertex equally, meaning that the aggregated result would be simply the sum of all the neighboring style node features. However, the results of such naïve approach are likely to be affected by the noisy style vertices, resulting in undesired artifacts.

To tackle this challenge, we apply an attention coefficient for each style vertex during message passing, which is learned in a data-driven manner. Given a centering content node v_c and its neighboring style nodes $\{v_s\}$ with the cardinality of k, the learned attention coefficients $w(v_c, v_s^j)$ between v_c and a specific

neighbor v_s^j can be computed as:

$$w(v_c, v_s^j) = \frac{\exp\left(\text{LeakyReLU}\left(W_a[W_b\mathcal{F}_c \| W_b\mathcal{F}_s^j]\right)\right)}{\sum_{m=1}^{k}\exp\left(\text{LeakyReLU}\left(W_a[W_b\mathcal{F}_c \| W_b\mathcal{F}_s^m]\right)\right)}, \tag{2}$$

where W represents the learnable matrix in linear transformation. $\|$ is the concatenation operation.

With such an attention-based aggregation manner, our stylization GNN can adaptively collect more significant information from the best-matching style patches, and meanwhile reduce the features from the less-matching noisy ones. Furthermore, we also apply a multi-headed architecture that generates the multi-head attention, so as to stabilize the attention learning process.

Content-to-Content Message Passing. With the updated node features at the content vertices from the preceding style-to-content message passing process, we also perform a second-phase information propagation among different content nodes. The rationale behind our content-to-content message passing is to perform global patch-based adjustment upon the results of the style-to-content stage, by considering the inter-relationship between the stylized patches at different locations. As such, the global coherence can be maintained, where the content objects that share similar semantics are more likely to resemble each other in stylization, which will be further validated in the experiments.

This proposed intra-content propagation also delivers the benefit of alleviating the artifacts resulting from potential style-content patch mismatching, by combining the features from the correctly-matching results. The detailed content-to-content message passing procedure is analogous to that in style-to-content message passing, but replacing the style vertices in Eq. 2 with the neighboring content vertices with the associated updated node features.

The eventual aggregation results from the proposed inter- and intra-domain message passing are then converted back into the feature patches by a `Feat2Patch` operation, which is an inverse operation of `Patch2Feat`. The obtained patches are further transformed into the feature map for the use of the subsequent global feature alignment module and feature decoding module.

3.4 Loss Function and Training Strategy

To align the semantic content, our content loss $\mathcal{L}_c$ is defined as the perceptual loss over the features from layer $\{\texttt{relu4\_1}\}$ of the pre-trained VGG network Φ:

$$\mathcal{L}_c = \|\Phi^{\texttt{relu4\_1}}(\mathcal{I}_c) - \Phi^{\texttt{relu4\_1}}(\mathcal{I}_o)\|_2, \tag{3}$$

where $\mathcal{I}_c$ and $\mathcal{I}_o$ represent the content and the output stylized images, respectively. For the style loss, we use the BN-statistic loss to extrat and transfer the style information, computed at layer $\{\texttt{relu1\_1}, \texttt{relu2\_1}, \texttt{relu3\_1}, \texttt{relu4\_1}\}$ of the VGG network Φ:

$$\mathcal{L}_s(h) = \sum_{\ell=1}^{4}\left(\left\|h\left(\Phi^{\texttt{relu}\ell\texttt{\_1}}(\mathcal{I}_s)\right) - h\left(\Phi^{\texttt{relu}\ell\texttt{\_1}}(\mathcal{I}_o)\right)\right\|_2\right), \tag{4}$$

Algorithm 1 Training a GNN-based stylization model that can transfer arbitrary styles in a semi-parametric manner.

Input: $\mathcal{I}_c$: the content image; $\mathcal{I}_s$: the style image; `VGG`: the pre-trained loss network.
Output: $\mathcal{I}_o$: Target stylized image that simultaneously preserves the appearance of $\mathcal{I}_s$ and the semantics of $\mathcal{I}_c$.

Perform initializations on the image encoder $\texttt{Enc}(\cdot)$, the scale predictor $\texttt{Prec}(\cdot)$, GNN parameters W_a and W_b, and the feature decoder $\texttt{Dec}(\cdot)$.
for $i = 1$ to $\mathcal{T}$ iterations **do**
    Feed $\mathcal{I}_s$ and $\mathcal{I}_c$ into $\texttt{Enc}(\cdot)$ and obtain the style and content features $\mathcal{F}_s$ and $\mathcal{F}_c$;
    Divide $\mathcal{F}_c$ into equal-size content patches $\{\phi(\mathcal{F}_c)\}$ by using a sliding window;
    Feed $\{\mathcal{F}_s, \mathcal{F}_c\}$ into $\texttt{Prec}(\cdot)$ and obtain the optimal scales $\{\alpha\}$ for style patches;
    Divide $\mathcal{F}_s$ into varying-size style patches $\{\phi(\mathcal{F}_s)\}$ with the obtained scales $\{\alpha\}$;
    Resize $\{\phi(\mathcal{F}_s)\}$ according to the size of the content patches $\{\phi(\mathcal{F}_c)\}$;
    Construct inter- and intra-domain edges by Eq. 1;
    Transform $\{\phi(\mathcal{F}_s)\}$ and $\{\phi(\mathcal{F}_c)\}$ into the node features by using `Patch2Feat`;
    Establish the heterogeneous graph $\mathcal{G}_{NST}$ and feed $\mathcal{G}_{NST}$ into the GNN layers;
    Perform heterogeneous message passing over $\mathcal{G}_{NST}$ by Eq. 2 and obtain $\mathtt{f_c}$;
    Convert the aggregation results $\mathtt{f_c}$ into feature map $\mathcal{F}_o$ by `Feat2Patch`;
    Feed the obtained features $\mathcal{F}_o$ into the global feature refiner and obtain $\mathcal{F}'_o$;
    Feed $\mathcal{F}'_o$ into the decoder $\texttt{Dec}(\cdot)$ to obtain the target stylized image $\mathcal{I}_o$;
    Feed $\{\mathcal{I}_o, \mathcal{I}_c, \mathcal{I}_s\}$ into `VGG` and compute $\mathcal{L}_c$ and $\mathcal{L}_s$ by Eq. 3 and Eq. 4;
    Optimize $\texttt{Enc}(\cdot)$, $\texttt{Prec}(\cdot)$, W_a, W_b, and $\texttt{Dec}(\cdot)$ with the Adam optimizer;
end for

where $h(\cdot)$ denotes the mapping of computing the BN statistics over the feature maps. The style loss can then be defined as: $\mathcal{L}_s = \mathcal{L}_s(\mu) + \mathcal{L}_s(\sigma)$, with $\mu(\cdot)$ and $\sigma(\cdot)$ denoting mean and standard standard deviation, respectively.

Our total loss is thereby a weighted sum of the content and style loss, formulated as: $\mathcal{L} = \mathcal{L}_{content} + \lambda\mathcal{L}_{style}$ with λ as the weighting factor that balances the content and style portions.

We also derive an elaborated training pipeline, tailored for the proposed GNN-based semi-parametric style transfer framework. As a whole, the detailed process of training a GNN-based semi-parametric arbitrary stylization model with the proposed algorithm is concluded in Alg. 1.

4 Experiments

4.1 Experimental Settings

We demonstrate here the implementation details as per the stage of the proposed semi-parametric pipeline. For the stylization graph construction stage, we set k as 5 by default for the NCC-based KNN search. The stride s for the sliding window is set to 1, whereas the kernel size is set to 5×5. At the stage of deformable graph convolution, we primarily use the graph attention network (GAT) [43] for the GNN layers to validate the effectiveness of the proposed semi-parametric NST scheme. During training, we adopt the Adam optimizer [23] to optimize the whole GNN-based network. The learning rate is 1×10^{-4} with a weight decay of 5×10^{-5}. The batch size is set to 8. The weighting factor

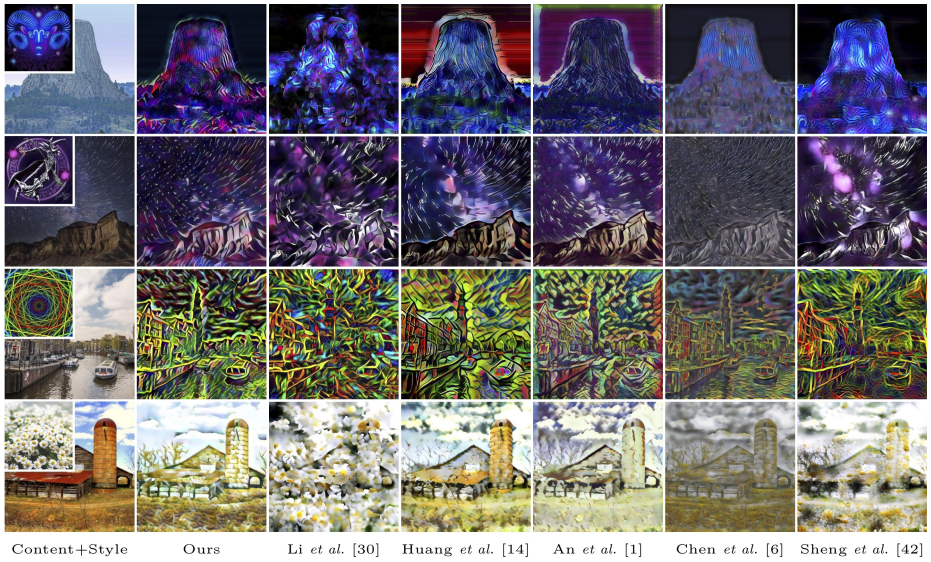

Fig. 3. Qualitative results of our proposed GNN-based semi-parametric stylization algorithm and other parametric [30, 14, 1] and non-parametric [6, 42] methods.

λ is set to 10. We employ a pre-trained VGG-19 as our loss network, as also done in [11, 14]. The network is trained on the Microsoft COCO dataset [32] and the WikiArt [38] dataset. Our code is based on Deep Graph Library (DGL) [44]. The training takes roughly two days on an NVIDIA Tesla A100 GPU.

4.2 Results

Qualitative comparison. Fig. 3 demonstrates the results of the proposed GNN-based semi-parametric method and other arbitrary style transfer methods [30, 14, 1, 6, 42]. The results of [30] are prone to distorted patterns. By contrast, the algorithms of [14, 1] generate sharper details; however, the local style patterns in their results are not well aligned with the target ones, where very few fine strokes are produced for most styles. For the non-parametric NST approaches of [6, 42], their stylized results either introduce fewer style patterns or suffer from artifacts, due to the potential issue of one-to-one patch mismatching. Compared with other approaches, our semi-parametric framework leads to few artifacts, and meanwhile preserves both the global style appearance and the local fine details, thanks to the local patch-based manipulation module with GNNs.

Efficiency analysis. In Tab. 1, we compare the average stylization speed of the proposed approach with other algorithms. For a fair comparison, all the methods are implemented with PyTorch. The experiments are performed over 100 equal-size content and style images of different resolutions using an NVIDIA Tesla A100 GPU. Our speed is, in fact, bottlenecked by the KNN search process, which can be further improved with an optimized KNN implementation.

Table 1. Average speed comparison in terms of seconds per image.

Methods	Time (s)		
	256×256	384×384	512×512
Li *et al.* [30]	0.707	0.779	0.878
Huang *et al.* [14]	0.007	0.010	0.017
An *et al.* [1]	0.069	0.108	0.169
Chen *et al.* [6]	0.017	0.051	0.218
Sheng *et al.* [42]	0.412	0.536	0.630
Ours	0.094	0.198	0.464

4.3 Ablation Studies

Heterogeneous aggregation schemes. We show in Fig. 4 the stylization results by using different neighborhood aggregation strategies in the local patch-based manipulation module. The results of the GAT aggregation scheme, as shown in the 3$^{\text{rd}}$ column of Fig. 4, outperform those of others in finer structures and global coherence (the areas of the sky and the human face in Fig. 4), thereby validating the superiority of the attention scheme in Eq. 2.

Stylization w/ and w/o the deformable scheme. Fig. 5 demonstrates the results with the equal-size patch division method, and those with the proposed deformable patch splitting scheme. The devised deformable module makes it possible to adaptively control the strokes in different areas. As a result, the contrast information in the stylized results can be enhanced.

Graph w/ and w/o intra-domain edges. In Fig. 6, we validate the effectiveness of the proposed content-to-content message passing scheme, which typically leads to more consistent style patterns in semantically-similar content regions, as can be observed in the foreground human and fox eye areas, as well as the background regions of Fig. 6.

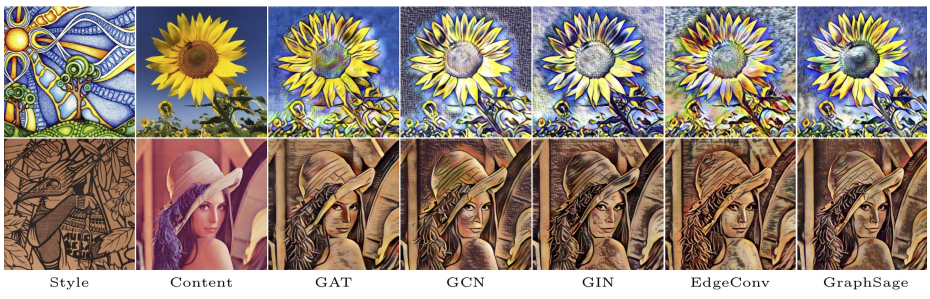

Fig. 4. Comparative results of using various aggregation mechanisms for heterogeneous message passing, including graph attention network (GAT) [43], graph convolutional network (GCN) [24], graph isomorphism network (GIN) [48], dynamic graph convolution (EdgeConv) [46], and GraphSage [12]. The GAT mechanism generally yields superior stylization results, thanks to its attention-based aggregation scheme in Eq. 2.

Fig. 5. Results of the equal-size patch division method and the proposed deformable one with a learnable scale predictor. Our deformable scheme allows for cross-scale style-content matching, thereby leading to spatially-adaptive multi-stroke stylization with an enhanced semantic saliency (*e.g.*, the foreground regions of the horse and squirrel).

Fig. 6. Results of removing the content-to-content intra-domain edges (w/o Intra) and those with the intra-domain ones (w/ Intra). The devised intra-domain connections incorporate the inter-relationship between the stylized patches at different locations, thereby maintaining the global stylization coherence (*e.g.*, the eye regions in the figure).

Euclidean distance *vs.* normalized cross-correlation. Fig. 7 shows the results of using the Euclidean distance and the normalized cross-correlation (NCC) as the distance metric, respectively, in the construction of the stylization graph. The adopted metric of NCC in our framework, as observed from the 4^{th} and 8^{th} columns of Fig. 7, leads to superior performance than the Euclidean distance (Fig. 7, the 3^{rd} and 7^{th} columns) in terms of both the global stroke arrangements and local details.

Various patch sizes. We show in Fig. 8 the results of diversified feature patch sizes. Larger patch sizes, as shown from the left to right in the figure, generally lead to larger strokes in the stylized results, which is especially obvious when we observe the regions of the dog and horse in Fig. 8.

4.4 Diversified Stylization Control

The proposed GNN-based arbitrary style transfer scheme, as shown in Fig. 9, can readily support diversified stylization with solely a single model. We also zoom in

Fig. 7. Results obtained using Euclidean distance and normalized cross-correlation (NCC) for similarity measurement during the construction of heterogeneous edges.

Fig. 8. Results obtained using various patch sizes for constructing content and style vertices in the local patch-based manipulation module. By using a larger patch size, the stylized results can maintain an overall larger stroke size.

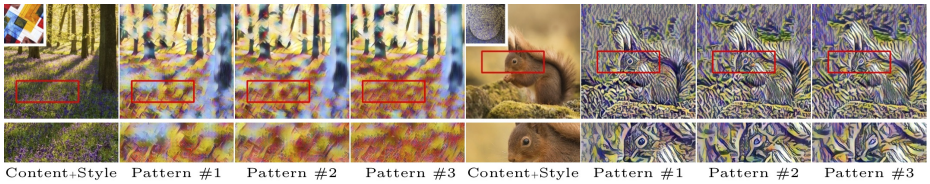

Fig. 9. Flexible control of diversified patch-based arbitrary style transfer during inference. The proposed GNN-based semi-parametric stylization scheme makes it possible to generate heterogeneous style patterns with only a single trained model.

on the same regions (*i.e.*, the red frames in Fig. 9) to observe the details. Such diversities in Fig. 9 are specifically achieved by simply changing the numbers of node-specific connections for heterogeneous message passing, which provide users of various tastes with more stylization choices.

5 Conclusions

In this paper, we introduce a semi-parametric arbitrary style transfer scheme for the effective transfers of challenging style patterns at the both local and global levels. Towards this goal, we identify two key challenges in existing parametric and non-parametric stylization approaches, and propose a dedicated GNN-based style transfer scheme to solve the dilemma. This is specifically accomplished by modeling the style transfers as the heterogeneous information propagation process among the constructed content and style vertices for accurate patch-based style-content correspondences. Moreover, we develop a deformable graph convolutional network for various-scale stroke generations. Experiments demonstrate that the proposed approach achieves favorable performance in both global stroke arrangement and local details. In our future work, we will strive to generalize the proposed GNN-based scheme to other vision tasks.

Acknowledgments. Mr Yongcheng Jing is supported by ARC FL-170100117. Dr Xinchao Wang is supported by AI Singapore (Award No.: AISG2-RP-2021-023) and NUS Faculty Research Committee Grant (WBS: A-0009440-00-00).

References

1. An, J., Huang, S., Song, Y., Dou, D., Liu, W., Luo, J.: Artflow: Unbiased image style transfer via reversible neural flows. In: CVPR (2021)
2. Champandard, A.J.: Semantic style transfer and turning two-bit doodles into fine artworks. arXiv preprint arXiv:1603.01768 (2016)
3. Chen, D., Yuan, L., Liao, J., Yu, N., Hua, G.: Stylebank: An explicit representation for neural image style transfer. In: CVPR (2017)
4. Chen, D., Yuan, L., Liao, J., Yu, N., Hua, G.: Explicit filterbank learning for neural image style transfer and image processing. TPAMI (2020)
5. Chen, H., Zhao, L., Zhang, H., Wang, Z., Zuo, Z., Li, A., Xing, W., Lu, D.: Diverse image style transfer via invertible cross-space mapping. In: ICCV (2021)
6. Chen, T.Q., Schmidt, M.: Fast patch-based style transfer of arbitrary style. In: NeurIPS Workshop on Constructive Machine Learning (2016)
7. Chen, Z., Zhu, Y., Zhao, C., Hu, G., Zeng, W., Wang, J., Tang, M.: Dpt: Deformable patch-based transformer for visual recognition. In: ACM MM (2021)
8. Ding, L., Wang, L., Liu, X., Wong, D.F., Tao, D., Tu, Z.: Understanding and improving lexical choice in non-autoregressive translation. In: ICLR (2021)
9. Ding, L., Wang, L., Tao, D.: Self-attention with cross-lingual position representation. In: ACL (2020)
10. Ding, L., Wang, L., Wu, D., Tao, D., Tu, Z.: Context-aware cross-attention for non-autoregressive translation. In: COLING (2020)
11. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: CVPR (2016)
12. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: NeurIPS (2017)
13. Hong, K., Jeon, S., Yang, H., Fu, J., Byun, H.: Domain-aware universal style transfer. In: ICCV (2021)
14. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: ICCV (2017)
15. Huo, J., Jin, S., Li, W., Wu, J., Lai, Y.K., Shi, Y., Gao, Y.: Manifold alignment for semantically aligned style transfer. In: ICCV (2021)
16. Jing, Y., Liu, X., Ding, Y., Wang, X., Ding, E., Song, M., Wen, S.: Dynamic instance normalization for arbitrary style transfer. In: AAAI (2020)
17. Jing, Y., Liu, Y., Yang, Y., Feng, Z., Yu, Y., Tao, D., Song, M.: Stroke controllable fast style transfer with adaptive receptive fields. In: ECCV (2018)
18. Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y., Song, M.: Neural style transfer: A review. TVCG (2019)
19. Jing, Y., Yang, Y., Wang, X., Song, M., Tao, D.: Amalgamating knowledge from heterogeneous graph neural networks. In: CVPR (2021)
20. Jing, Y., Yang, Y., Wang, X., Song, M., Tao, D.: Meta-aggregator: Learning to aggregate for 1-bit graph neural networks. In: ICCV (2021)
21. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV (2016)
22. Kalischek, N., Wegner, J.D., Schindler, K.: In the light of feature distributions: moment matching for neural style transfer. In: CVPR (2021)
23. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
24. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)

25. Kolkin, N., Salavon, J., Shakhnarovich, G.: Style transfer by relaxed optimal transport and self-similarity. In: CVPR (2019)
26. Kong, Y., Liu, L., Wang, J., Tao, D.: Adaptive curriculum learning. In: ICCV (2021)
27. Li, C., Wand, M.: Combining markov random fields and convolutional neural networks for image synthesis. In: CVPR. pp. 2479–2486 (2016)
28. Li, Y., Wang, N., Liu, J., Hou, X.: Demystifying neural style transfer. In: IJCAI (2017)
29. Li, Y., Chen, F., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Diversified texture synthesis with feed-forward networks. In: CVPR (2017)
30. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Universal style transfer via feature transforms. In: NeurIPS (2017)
31. Liao, J., Yao, Y., Yuan, L., Hua, G., Kang, S.B.: Visual attribute transfer through deep image analogy. TOG (2017)
32. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
33. Liu, H., Yang, Y., Wang, X.: Overcoming catastrophic forgetting in graph neural networks. In: AAAI (2021)
34. Liu, S., Lin, T., He, D., Li, F., Deng, R., Li, X., Ding, E., Wang, H.: Paint transformer: Feed forward neural painting with stroke prediction. In: ICCV (2021)
35. Liu, S., Lin, T., He, D., Li, F., Wang, M., Li, X., Sun, Z., Li, Q., Ding, E.: Adaattn: Revisit attention mechanism in arbitrary neural style transfer. In: ICCV (2021)
36. Liu, X.C., Yang, Y.L., Hall, P.: Learning to warp for style transfer. In: CVPR (2021)
37. Mechrez, R., Talmi, I., Zelnik-Manor, L.: The contextual loss for image transformation with non-aligned data. In: ECCV (2018)
38. Nichol, K.: Painter by numbers (2016), https://www.kaggle.com/c/painter-by-numbers
39. Ren, S., Zhou, D., He, S., Feng, J., Wang, X.: Shunted self-attention via multi-scale token aggregation. In: CVPR (2022)
40. Risser, E., Wilmot, P., Barnes, C.: Stable and controllable neural texture synthesis and style transfer using histogram losses. arXiv preprint arXiv:1701.08893 (2017)
41. Shen, C., Yin, Y., Wang, X., Li, X., Song, J., Song, M.: Training generative adversarial networks in one stage. In: CVPR (2021)
42. Sheng, L., Shao, J., Lin, Z., Warfield, S., Wang, X.: Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In: CVPR (2018)
43. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. In: ICLR (2018)
44. Wang, M., Yu, L., Zheng, D., Gan, Q., Gai, Y., Ye, Z., Li, M., Zhou, J., Huang, Q., Ma, C., et al.: Deep graph library: Towards efficient and scalable deep learning on graphs. In: ICLR Workshop (2019)
45. Wang, P., Li, Y., Vasconcelos, N.: Rethinking and improving the robustness of image style transfer. In: CVPR (2021)
46. 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)
47. Wu, X., Hu, Z., Sheng, L., Xu, D.: Styleformer: Real-time arbitrary style transfer via parametric style composition. In: ICCV (2021)
48. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: ICLR (2019)
49. Xu, W., Long, C., Wang, R., Wang, G.: Drb-gan: A dynamic resblock generative adversarial network for artistic style transfer. In: ICCV (2021)

50. Xu, Y., Zhang, Q., Zhang, J., Tao, D.: Vitae: Vision transformer advanced by exploring intrinsic inductive bias. NeurIPS (2021)
51. Yang, Y., Feng, Z., Song, M., Wang, X.: Factorizable graph convolutional networks. NeurIPS (2020)
52. Yang, Y., Qiu, J., Song, M., Tao, D., Wang, X.: Distilling knowledge from graph convolutional networks. In: CVPR (2020)
53. Yang, Y., Ren, Z., Li, H., Zhou, C., Wang, X., Hua, G.: Learning dynamics via graph neural networks for human pose estimation and tracking. In: CVPR (2021)
54. Yang, Y., Wang, X., Song, M., Yuan, J., Tao, D.: Spagan: Shortest path graph attention network. In: IJCAI (2019)
55. Ye, J., Jing, Y., Wang, X., Ou, K., Tao, D., Song, M.: Edge-sensitive human cutout with hierarchical granularity and loopy matting guidance. TIP (2019)
56. Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., Yan, S.: Metaformer is actually what you need for vision. In: CVPR (2022)
57. Zhan, Y., Yu, J., Yu, T., Tao, D.: On exploring undetermined relationships for visual relationship detection. In: CVPR (2019)
58. Zhan, Y., Yu, J., Yu, T., Tao, D.: Multi-task compositional network for visual relationship detection. IJCV (2020)
59. Zhang, H., Dana, K.: Multi-style generative network for real-time transfer. arXiv preprint arXiv:1703.06953 (2017)
60. Zhang, Q., Xu, Y., Zhang, J., Tao, D.: Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond. arXiv preprint arXiv:2202.10108 (2022)
61. Zhang, Q., Xu, Y., Zhang, J., Tao, D.: Vsa: Learning varied-size window attention in vision transformers. arXiv preprint arXiv:2204.08446 (2022)
62. Zhao, H., Bian, W., Yuan, B., Tao, D.: Collaborative learning of depth estimation, visual odometry and camera relocalization from monocular videos. In: IJCAI (2020)
63. Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: A review of methods and applications. arXiv preprint arXiv:1812.08434 (2018)
64. Zhou, S., Zhang, J., Zuo, W., Loy, C.C.: Cross-scale internal graph neural network for image super-resolution. NeurIPS (2020)