Texture Reformer: Towards Fast and Universal Interactive Texture Transfer

Zhizhong Wang, Lei Zhao*, Haibo Chen, Ailin Li, Zhiwen Zuo, Wei Xing*, Dongming Lu

College of Computer Science and Technology, Zhejiang University
{endywon, cszhl, cshbchen, liailin, zzwcs, wxing, ldm}@zju.edu.cn

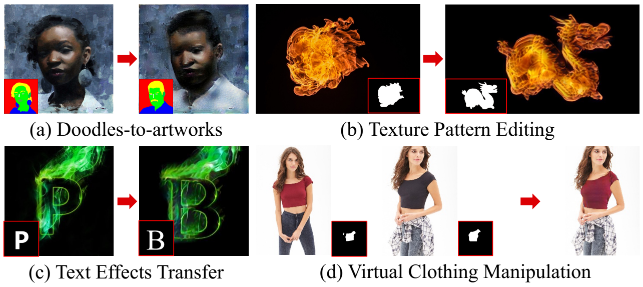
Abstract

In this paper, we present the texture reformer, a fast and universal neural-based framework for interactive texture transfer with user-specified guidance. The challenges lie in three aspects: 1) the diversity of tasks, 2) the simplicity of guidance maps, and 3) the execution efficiency. To address these challenges, our key idea is to use a novel feed-forward multi-view and multi-stage synthesis procedure consisting of I) a global view structure alignment stage, II) a local view texture refinement stage, and III) a holistic effect enhancement stage to synthesize high-quality results with coherent structures and fine texture details in a coarse-to-fine fashion. In addition, we also introduce a novel learning-free view-specific texture reformation (VSTR) operation with a new semantic map guidance strategy to achieve more accurate semantic-guided and structure-preserved texture transfer. The experimental results on a variety of application scenarios demonstrate the effectiveness and superiority of our framework. And compared with the state-of-the-art interactive texture transfer algorithms, it not only achieves higher quality results but, more remarkably, also is 2-5 orders of magnitude faster.

1 Introduction

As a variant of texture synthesis, texture transfer is a long-standing problem that seeks to transfer the stylized texture from a given sample to the target image (Efros and Freeman 2001). After the rapid development in recent years, a bunch of conventional (Hertzmann et al. 2001) or neural-based (Gatys, Ecker, and Bethge 2016) methods have been proposed and obtained visually appealing results. However, due to the lack of user guidance, general texture transfer methods often produce unsatisfying results against human expectations. To resolve this dilemma, the community resorts to using the user-specified semantic maps to guide the transfer process, which is called *interactive texture transfer* (Men et al. 2018). Users can control the shape, scale, and spatial distribution of the objects to be synthesized in the target image via semantic maps.

At first, the interactive texture transfer methods are only designed for specific usage scenarios. (Champandard 2016) proposed Neural Doodle to turn doodles painted by users into fine artworks with provided samples. Lu *et al.* designed HelpingHand (Lu et al. 2012), RealBrush (Lu et al. 2013), and DecoBrush (Lu et al. 2014) to edit different kinds of texture patterns. (Yang et al. 2017, 2019) achieved text effects transfer that can migrate fantastic text effects of stylized texts onto raw plain texts. (Han et al. 2018) introduced a virtual try-on network to transfer a target clothing item to the corresponding region of a clothed person. These approaches seem to be isolated, but they all share a common notion of transferring textures under user guidance.



(a) Doodles-to-artworks (b) Texture Pattern Editing

(c) Text Effects Transfer (d) Virtual Clothing Manipulation

Figure 1: Representative results generated by our interactive texture reformer. The stylized images are synthesized under the guidance of corresponding user-specified semantic maps. Our framework is universal for multiple challenging user-controlled texture transfer tasks, *e.g.*, (a) turning doodles into artworks, (b) editing texture patterns, (c) transferring text effects, (d) manipulating clothing textures and distributions. Compared with the state-of-the-art interactive texture transfer algorithms, *it not only can achieve higher quality results but, more remarkably, also is 2-5 orders of magnitude faster.*

To unify them, (Men et al. 2018) proposed a common framework for interactive texture transfer by incorporating multiple custom channels to dynamically guide the synthesis. This method can handle various tasks and achieves the state of the art. However, it relies on several CPU-based operations and a backward optimization process, thus usually requiring several minutes to generate a result for each interaction, which is prohibitively slow. Therefore, existing algorithms are hard to satisfy the practical requirements due to the limitations of efficiency or application scenarios. A fast

*Corresponding authors.

and universal framework is eagerly desired, and it will undoubtedly improve the user experience and bring higher application and research value to both industry and academia.

However, achieving such a goal is a rather challenging task. The challenges mainly lie in three aspects: 1) The diversity of tasks: the discrepancies between different tasks make the transfer problem difficult to model uniformly. Besides, for each task, the algorithm should be robust to different input samples. 2) The simplicity of guidance maps: the doodle semantic map as guidance gives few hints on how to place different inner textures and preserve local high-frequency structures (Men et al. 2018). 3) The execution efficiency: the trade-off between efficiency and quality is always an intractable problem. This is particularly important for interactive systems since the insufficient computational speed not only brings inconvenience to users but also hampers the truly exploratory use of these techniques.

To address these challenges, in this paper, we propose the *texture reformer*, a fast and universal neural-based framework for interactive texture transfer with user-specified guidance. The key insight is to use a novel feed-forward multi-view and multi-stage synthesis procedure, which consists of three different stages: I) a global view structure alignment stage, II) a local view texture refinement stage, and III) a holistic effect enhancement stage. Specifically, for stage I and II, we introduce a novel *View-Specific Texture Reformation (VSTR)* operation with a new semantic map guidance strategy to achieve more accurate semantic-guided and structure-preserved texture transfer. By specifying a global view for VSTR, our stage I first captures and aligns the inner structures of the source textures as completely as possible. Then, the results are carefully rectified and refined in stage II via specifying a local view for VSTR. Finally, in stage III, we leverage the Statistics-based Enhancement (SE) operations to further enhance the low-level holistic effects (*e.g.*, colors, brightness, and contrast). Note that our framework is built upon several auto-encoder networks trained solely for image reconstruction, and the VSTR and SE operations are *learning-free*. Therefore, it can achieve interactive texture transfer universally. By cascading the above three stages, our texture reformer can synthesize high-quality results with coherent structures and fine texture details in a coarse-to-fine fashion. We demonstrate the effectiveness and superiority of our framework on a variety of application scenarios, including doodles-to-artworks, texture pattern editing, text effects transfer, and virtual clothing manipulation (see Fig. 1). The experimental results show that compared with the state-of-the-art algorithms, our texture reformer not only achieves higher quality results but, more remarkably, also is 2-5 orders of magnitude faster. *As far as we know, our work is the first to meet the requirements of quality, flexibility, and efficiency at the same time in this task.*

In summary, our contributions are threefold:

- We propose a novel multi-view and multi-stage neural-based framework, *i.e.*, *texture reformer*, to achieve fast and universal interactive texture transfer for the first time.
- We also introduce a novel learning-free *view-specific texture reformation (VSTR)* operation with a new semantic map guidance strategy, to realize more accurate semantic-guided and structure-preserved texture transfer.
- We apply our framework to many challenging interactive texture transfer tasks, and demonstrate its effectiveness and superiority through extensive comparisons with the state-of-the-art (SOTA) algorithms.

2 Related Work

Conventional Texture Transfer. Conventional texture transfer relies on hand-crafted algorithms (Haeberli 1990) or features (Kwatra et al. 2005) to migrate the textures from source samples to target images. The pioneering works of (Efros and Leung 1999; Efros and Freeman 2001) sampled similar patches to synthesize and transfer textures. Later, (Hertzmann et al. 2001) proposed Image Analogy to generate the stylized result of the target image. (Barnes et al. 2009, 2010) proposed PatchMatch to accelerate the nearest-neighbor search process, which was further extended to image melding (Darabi et al. 2012), style transfer (Frigo et al. 2016), and text effects transfer (Yang et al. 2017), *etc*. However, for interactive texture transfer, these methods fail to synthesize textures with salient structures and are prone to wash-out effects (Men et al. 2018). To combat the issues, (Men et al. 2018) proposed a common framework for interactive texture transfer by utilizing an improved PatchMatch and multiple custom channels to dynamically guide the synthesis, achieving SOTA performance. However, as analyzed in Sec. 1, this method suffers from rather slow computational speed, thus cannot satisfy the practical requirements.

Unlike the SOTA conventional texture transfer methods (Yang et al. 2017; Men et al. 2018), our proposed texture reformer is neural-based, and not only can achieve higher quality results but also is several orders of magnitude faster.

Neural-based Style Transfer. The seminal works of (Gatys, Ecker, and Bethge 2016, 2015) have proved the power of Deep Convolutional Neural Networks (DCNNs) (Simonyan and Zisserman 2014) in style transfer and texture synthesis, where the Gram matrices of the features extracted from different layers of DCNNs are used to represent the style of images. Further works improved it in many aspects, including efficiency (Johnson, Alahi, and Fei-Fei 2016), quality (Jing et al. 2018; Kolkin, Salavon, and Shakhnarovich 2019; Park and Lee 2019; Wang et al. 2020b, 2021; Chen et al. 2020, 2021b,a; An et al. 2021), generality (Li et al. 2017; Huang and Belongie 2017; Zhang, Zhu, and Zhu 2019; Jing et al. 2020), and diversity (Wang et al. 2020a; Chen et al. 2021c). For interactive style transfer, (Gatys et al. 2017) introduced user spatial control into (Gatys, Ecker, and Bethge 2016), which is further accelerated by (Lu et al. 2017). However, due to the characteristics of Gram matrix matching, these methods often produce disordered textures, which cannot preserve the local inner structures, as will be demonstrated in later Sec. 4.3.

Another line of neural-based style transfer is based on neural patches. (Li and Wand 2016a,b) first achieved it by combining Markov Random Fields (MRFs) and DCNNs. (Liao et al. 2017) proposed Deep Image Analogy for more accurate semantic-level patch matching. Later, (Chen and

Schmidt 2016) leveraged a “style swap” operation for fast patch-based stylization. To incorporate user control, (Champandard 2016) augmented (Li and Wand 2016a) with semantic annotations, leading to higher quality and avoiding common glitches. However, the results usually contain too many low-level noises. Also, efficiency is concerned as it still relies on a time-consuming backward optimization process.

Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) provide another idea to generate textures by training discriminator and generator networks to play an adversarial game. cGANs (Mirza and Osindero 2014), which were further extended by (Isola et al. 2017), have been applied to many image manipulation tasks, such as image editing (Zhu et al. 2016), texture synthesis (Frühstück, Alhashim, and Wonka 2019), sketch2image (Chen and Hays 2018), and inpainting (Zhao et al. 2020), *etc*. However, all of them are trained on class-specific datasets. By contrast, our method only needs one exemplar for generating the target image from a corresponding semantic map. Recently, some single image generative models (Shaham, Dekel, and Michaeli 2019; Lin et al. 2020) were also proposed to generate images based on only a single image. Nevertheless, these methods often produce poor results for images with complex texture details and structures, and need several hours to train the model on each pair of image samples.

3 Proposed Approach

We first describe the task of interactive texture transfer following the definitions in (Men et al. 2018). As illustrated in the left part of Fig. 3, given a stylized source image S_{sty} and its corresponding semantic map S_{sem}, interactive texture transfer aims to generate the stylized target image T_{sty} with a user-specified target semantic map T_{sem}. Users can control the shape, scale, and spatial distribution of the objects to be synthesized in the target image via semantic maps.

Using a semantic map that contains few hints to reproduce the structural image is a challenging task. The key challenge is to preserve the structures of the inner textures of each semantic region, *e.g.*, the clothing structures in the blue region of Fig. 3. (Men et al. 2018) combat it by introducing structure guidance based on the boundary patches of semantic maps to provide a prior in the synthesis procedure. However, it involves several structure extraction (Goferman, Zelnik-Manor, and Tal 2011) and propagation (Myronenko and Song 2010; Bookstein 1989) processes, which are cumbersome and time-consuming. In a fundamentally different way, we do not use any additional structure guidance but only benefit from the strong representative power of DCNNs to extract the multi-level image features. Based on these features, our key insight is to use a multi-view and multi-stage synthesis procedure to progressively generate structural textures in a coarse-to-fine fashion. In the following sections, we will first depict the overall pipeline and some critical components of our framework (Sec. 3.1), and then introduce each of its stages in detail (Sec. 3.2-3.4).

3.1 Overview of Texture Reformer

The overall pipeline of our framework is depicted in Fig. 2, which consists of three stages: I) a global view structure alignment stage, II) a local view texture refinement stage, and III) a holistic effect enhancement stage. Specifically, stage I is similar to a global copy-and-paste, which roughly aligns the spatial positions of the source patterns in S_{sty} to the target positions in the target semantic map T_{sem}. This global view alignment can help preserve the inner structures of the source patterns as completely as possible, which is critical to synthesize the structure-preserved textures. The warping and finer alignment is achieved via stage II, which uses a rather small local view, and can rectify and refine the results of stage I to a large extent, thus robust for different deformation requirements. Finally, the low-level holistic effects (*e.g.*, colors, brightness, and contrast) are further enhanced in stage III, thereby obtaining high-quality results. The visualizations of the inputs/outputs of each stage are shown in Fig. 2 (T^5_{sty}-T_{sty}). These stages are carried out at different levels of VGG (Simonyan and Zisserman 2014) features and are hierarchically cascaded to work in a coarse-to-fine fashion. Uniformly, they share the same workflow that generates the outputs using an AE (auto-encoder)-based image reconstruction process coupled with bottleneck feature operations. We adopt view-specific texture reformation (VSTR) for stage I and II, and statistics-based enhancement (SE) for stage III, which we will introduce in detail.

AE-based Image Reconstruction. We construct auto-encoder networks for general image reconstruction. We employ the first parts (up to $Relu\mathbf{X}\_1$) of a pre-trained VGG-19 (Simonyan and Zisserman 2014) as encoders, *fix* them and train symmetrical decoder networks with the nearest neighbor interpolation as upsampling layers for inverting the bottleneck features to the original RGB images. As shown in Fig. 2, in our framework, we select feature maps at five layers, *i.e.*, $Relu\mathbf{X}\_1$ (**X**=1,2,3,4,5), and train five decoders accordingly with the following loss:

$$\mathcal{L}_{recon} = \parallel I_r - I_i \parallel_2^2 + \lambda \parallel \Phi(I_r) - \Phi(I_i) \parallel_2^2, \quad (1)$$

where I_i and I_r are the input image and reconstructed output, and Φ is the VGG encoder that extracts the $Relu\mathbf{X}\_1$ features. The decoders are trained on the Microsoft COCO dataset (Lin et al. 2014) and λ is set to 1.

View-Specific Texture Reformation (VSTR). We propose a novel *learning-free* VSTR operation to robustly propagate the thorough texture patterns onto the target features under the guidance of semantic maps. Denote $F^{S_{sty}}$ and $F^{T^t_{sty}}$ as the VGG features (*e.g.*, extracted from $Relu5\_1$) of the stylized source image S_{sty} and the *temporary* stylized target image T^t_{sty}. We first project them into a common space to standardize the data and dispel the domain gap,

$$F_1^{S_{sty}} = \frac{F^{S_{sty}} - \mu(F^{S_{sty}})}{\sigma(F^{S_{sty}})}; \; F_1^{T^t_{sty}} = \frac{F^{T^t_{sty}} - \mu(F^{T^t_{sty}})}{\sigma(F^{T^t_{sty}})}, \quad (2)$$

where μ and σ are the mean and standard deviation.

Then, we fuse the information from the source and target semantic maps S_{sem} and T_{sem} to guide the propagations between corresponding semantic regions. Existing works (Champandard 2016; Gatys et al. 2017) often directly concatenate the downsampled semantic maps, like follows:

$$F^S = F_1^{S_{sty}} \parallel \omega S^l_{sem}; \; F^T = F_1^{T^t_{sty}} \parallel \omega T^l_{sem}, \quad (3)$$

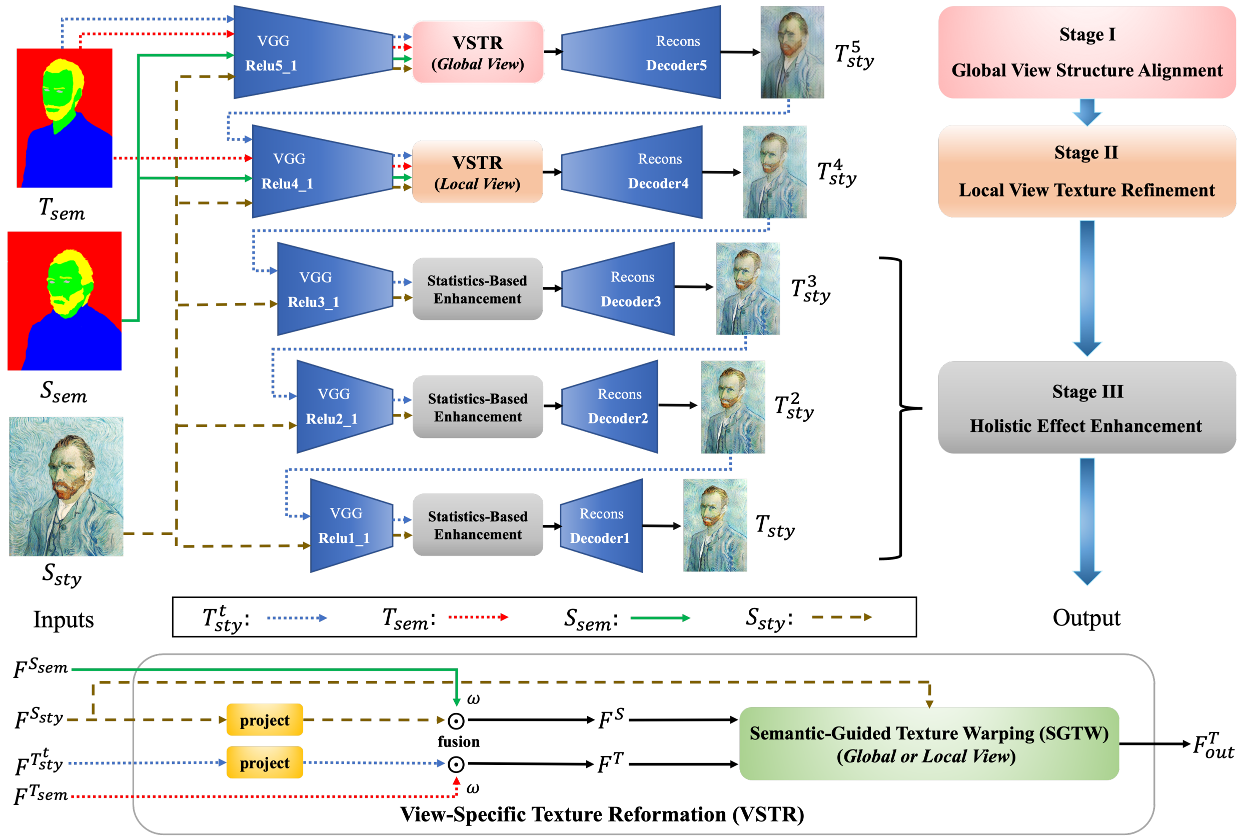
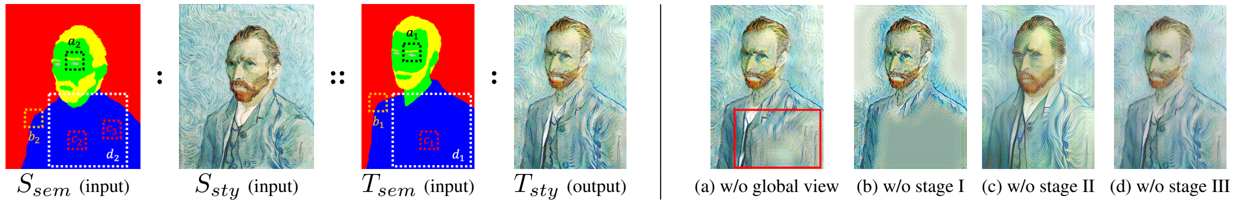


Figure 2: Overall pipeline of our proposed multi-view and multi-stage texture reformer.



	T_{sty} (output)	(a) w/o global view	(b) w/o stage I	(c) w/o stage II	(d) w/o stage III
Time/sec (averaged on 100 pairs of 512×512px images):	0.956	0.912	0.581	0.569	0.653

Figure 3: Left: Illustration of the interactive texture transfer task. Input three images: S_{sem} (semantic map of source image), S_{sty} (stylized source image aligned to S_{sem}), and T_{sem} (user-specified semantic map of target image), the stylized target image T_{sty} with the style of source image S_{sty} can be automatically synthesized such that $S_{sem} : S_{sty} :: T_{sem} : T_{sty}$. Right: Effects of different critical components and stages in our texture reformer (Fig. 2). Bottom: Efficiency comparison.

where $||$ denotes channel-wise concatenation. l denotes the downsampling factor. ω is the hyperparameter that controls the weight of semantic awareness. However, as pointed out by (Gatys et al. 2017), this method has limited capacity to model complex textures and usually produces inaccurate semantic matching (*e.g.*, the 2^{nd} column in Fig. 4). This can be attributed to the information discrepancy between the RGB images and deep VGG features. In addition, the difference in the amount of channels may also make it hard to find a good compromise (*i.e.*, the proper value of ω) between them.

To resolve this issue, we introduce a new semantic map guidance strategy in our VSTR. That is, first extracting the VGG features $F^{S_{sem}}$ and $F^{T_{sem}}$ for semantic maps S_{sem} and T_{sem}, and then conducting the fusion in the VGG embedding space.

$$F^S = F_1^{S_{sty}} \odot \omega F^{S_{sem}};\ F^T = F_1^{T^t_{sty}} \odot \omega F^{T_{sem}}, \qquad (4)$$

where the fusion operation $\odot$ can be channel-wise concatenation or position-wise addition. We find these two operations could perform closely in some cases (*e.g.*, the 3^{rd} and 4^{th} top images in Fig. 4). But in general, concatenation often achieves more accurate semantic guidance (see the 3^{rd} and 4^{th} bottom images in Fig. 4) yet addition can provide faster speed (see the bottom efficiency comparison in Fig. 4).

After obtaining the fused features F^S and F^T, inspired by (Chen and Schmidt 2016), we introduce a *Semantic-Guided*

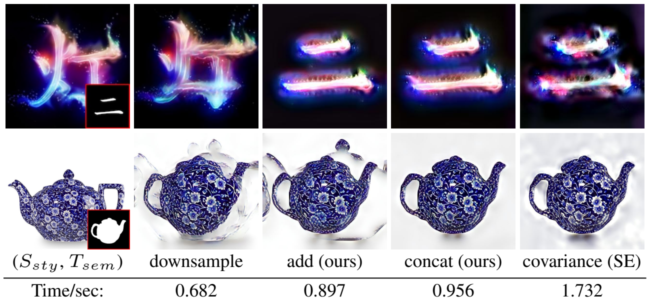


Figure 4: Comparison of different semantic map guidance strategies (2^{nd} to 4^{th} columns) and different enhancement operations (last column).

Texture Warping (SGTW) module with a specific field of view (*i.e.*, patch size p) to warp and transfer the textures. The detailed procedure is as follows:

1. Extract a set of $p \times p$ *original* source patches from the *original* source feature $F^{S_{sty}}$, denoted by $\{\phi_i(F^{S_{sty}})\}_{i \in \{1,\dots,n_s\}}$, where n_s is the number of extracted patches.
2. Extract a set of $p \times p$ *fused* source patches from the *fused* source feature F^S, denoted by $\{\phi_i(F^S)\}_{i \in \{1,\dots,n_s\}}$.
3. Determine the closest-matching *fused* source patch for each *fused* target patch in F^T by using a convolutional layer with the normalized fused source patches $\{\phi_i(F^S)/ \parallel \phi_i(F^S) \parallel\}$ as filters and F^T as input. The computed result $\mathcal{T}$ has n_s feature channels, and each spatial location is a cosine similarity vector between a fused target patch and all fused source patches.
4. Binarizing the scores in $\mathcal{T}$ such that the maximum value along the channel is **1** and the rest are **0**. The result is denoted as $\hat{\mathcal{T}}$.
5. Generate the output F^T_{out} by a deconvolutional layer with the *original* source patches $\{\phi_i(F^{S_{sty}})\}$ as filters and $\hat{\mathcal{T}}$ as input.

The novel insight behind SGTW is that we exploit the semantic-guided matching relationship between the patches of *fused* features F^S and F^T to reassemble and warp the *original source* feature $F^{S_{sty}}$. This not only guarantees the accurate alignment with the target semantic map, but also theoretically ensures that the output feature F^T_{out} can preserve the texture details of the original source feature $F^{S_{sty}}$ *losslessly*, since all its patches are from $F^{S_{sty}}$. Moreover, by specifying different views for SGTW, we can control the granularity of preserved texture details (*e.g.*, the integrity of inner structures) and the alignment accuracy with the target semantic map, as will be shown in later Sec. 3.2 and 3.3. By leveraging SGTW, our VSTR thus can realize more accurate semantic-guided and structure-preserved texture transfer. *Note that the matching and reassembling steps actually only add two convolutional layers to the feed-forward networks, and thus their implementation is very efficient.*

Statistics-based Enhancement (SE). This operation aims to enhance the holistic effects of the stylized target image based on global statistics matching. Either the first-order statistics (*e.g.*, mean and standard deviation) (Huang and Belongie 2017) or the second-order statistics (*e.g.*, covariance) (Li et al. 2017) can be adopted. In practice, we find the first-order statistics can work better in our task. As shown in the last column of Fig. 4, though higher-order statistics can reproduce the surface gloss of ceramic teapot more faithfully, they may produce inferior results with hazy shadows and consume much more time. Thus, we define our SE operation as a simple first-order statistics matching.

$$\begin{aligned} &SE(F^{S_{sty}}, F^{T^t_{sty}}) = \\ &\sigma(F^{S_{sty}})\left(\frac{F^{T^t_{sty}} - \mu(F^{T^t_{sty}})}{\sigma(F^{T^t_{sty}})}\right) + \mu(F^{S_{sty}}), \end{aligned} \tag{5}$$

where μ and σ are the mean and standard deviation.

3.2 Global View Structure Alignment Stage

As introduced in Sec. 3.1, the goal of this stage is to preserve the inner structures of the source textures as completely as possible so as to provide good structure guidance for subsequent stages. An important intuition we will use is that the inner structures with different scales can be captured from different views, and if we process from the global view, then the complete inner structures can be captured. For example, as we plotted in the semantic maps of Fig. 3, if we match T_{sem} and S_{sem} from a local view (*i.e.*, use a small patch size p), only the patches covering small or boundary structures in T_{sem} (*e.g.*, patch a_1 and b_1) can find the proper counterparts in S_{sem} (patch a_2 and b_2). For those in the large plain regions (*e.g.*, patch c_1), it is hard to choose their best-suited partners among internal source patches (*e.g.*, patch c_2 and c_3), since they are completely identical (both full-blue). Thus, the inner structures in these regions cannot be retained, and the results would show severe wash-out effects, like image (b) in the right part of Fig. 3 (where only the local view of stage II (Sec. 3.3) is used). However, if we enlarge the view of these hard patches to include some salient structures (*e.g.*, patch d_1), they can easily find the proper counterparts in S_{sem} again (patch d_2). At this point, the complete inner structures can be well captured and preserved.

Following this intuition, we make a global view setting in the VSTR of this stage to handle the feature maps from the global view, *i.e.*, using a dynamic global/maximum patch size p to cover the inner structures as completely as possible:

$$p = min[H(F^S), W(F^S), H(F^T), W(F^T)] - 1, \tag{6}$$

where H and W denote the height and width of the features. Unfortunately, operating directly on these large patches will severely grow the computation and time cost. To alleviate this issue, we resort to the deepest layer (*i.e.*, $Relu5\_1$) of VGG-19 to implement the global alignment. It brings two merits: (1) The costs can be minimized, as the features at this layer have the smallest size (see the efficiency comparison below T_{sty} and image (a) in Fig. 3). (2) This layer provides the highest-level structure features and the largest receptive field, perfectly suitable for this stage. For validation, we use a small local view (*i.e.*, $p = 3$) in this stage to obtain the right image (a) of Fig. 3. As observed in the red rectangle area, the result still suffers from wash-out effects that lose the inner structures, but here the effects are alleviated to some extent compared to the right image (b) (which is aligned at $Relu4\_1$). It indicates that our global view setting can help capture more intact inner structures, and the deeper VGG layer can provide higher-level structure features and larger receptive fields for better global alignment.

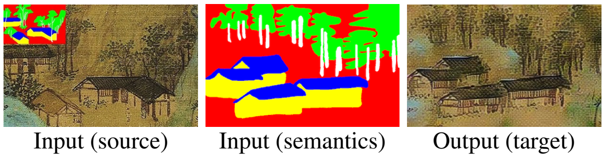


Figure 5: Doodles-to-artworks.

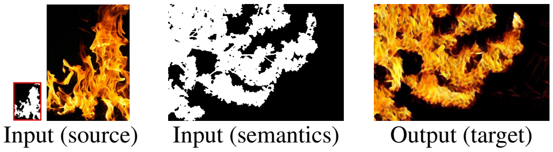


Figure 6: Texture Pattern Editing.

3.3 Local View Texture Refinement Stage

This stage takes the output of stage I as the input temporary stylized target image T^t_{sty} to guide a more detailed synthesis. Similar to stage I, it also uses VSTR to process the bottleneck features. The difference is, we process the features at a relatively shallower layer $Relu4\_1$, and use a much smaller patch size (*i.e.*, $p = 3$) to handle the features from only the local view. The effect of this stage can be inferred by comparing T_{sty} with the right image (c) in Fig. 3. The local view helps rectify and refine the local structures and texture details to a large extent, thus achieving more accurate alignment and higher quality.

3.4 Holistic Effect Enhancement Stage

The former two stages have been able to transfer satisfying inner structures and texture details. However, as they are based on high-level features, the synthesized images often neglect the low-level holistic effects (*e.g.*, colors, brightness, and contrast), as shown in the right image (d) of Fig. 3. To further enhance these low-level effects, this stage utilizes the statistics-based enhancement (SE) on the low-level features at three shallow layers, *i.e.*, $Relu\mathbf{X}\_1$ (**X**=1,2,3). As such, we can finally synthesize high-quality results which perform well in both high-level structures and low-level effects. Note that though our VSTR can also be used here to enhance the low-level effects, we do not recommend it as it will severely increase the time cost and memory requirement.

4 Experimental Results

4.1 Implementation Details

We adopt concatenation as the default setting to fuse semantic guidance. The hyperparameters that control the semantic-awareness (Eq. 4) in stage I and stage II are set to $\omega_1 = \omega_2 = 50$ (ω_1 for stage I, ω_2 for stage II. See *supplementary material (SM)* for their effects). Code is available at https://github.com/EndyWon/Texture-Reformer.

4.2 Applications

Our framework can be effectively applied to multiple interactive texture transfer tasks, such as doodles-to-artworks, texture pattern editing, text effects transfer, and virtual clothing manipulation (see the examples in Fig. 1, 5, 6, 7, 8).

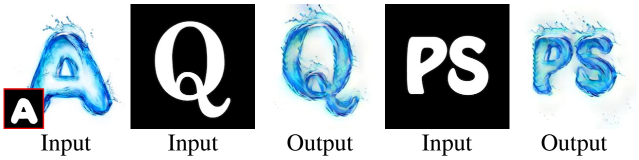
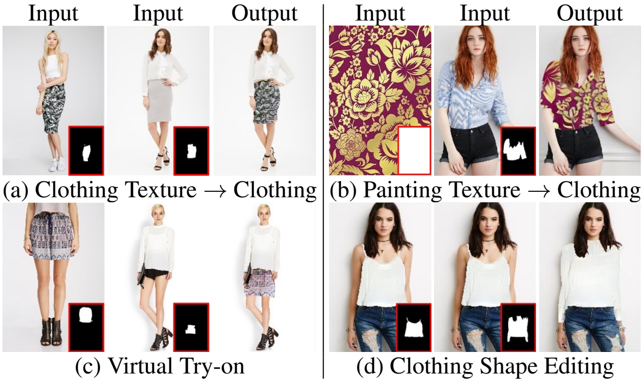


Figure 7: Text Effects Transfer.



(a) Clothing Texture → Clothing (b) Painting Texture → Clothing

(c) Virtual Try-on (d) Clothing Shape Editing

Figure 8: Virtual Clothing Manipulation.

Doodles-to-artworks. This task aims to turn the two-bit doodles annotated by users into fine artworks with similar styles as the given exemplar paintings or photographs, as shown in Fig. 5. See more in *SM*.

Texture Pattern Editing. As illustrated in Fig. 6, given an exemplar image, users can edit the texture patterns such as path and shape according to their needs. This provides a controllable way to modify the existing patterns.

Text Effects Transfer. As shown in Fig. 7, our method is also effective for text effects transfer which can migrate the artistic effects of stylized text images or source styles onto arbitrary raw plain texts.

Virtual Clothing Manipulation. Manipulating the clothing textures and distributions in a virtual way is an interesting and practical problem that has attracted much attention in recent years (Han et al. 2018, 2019a,b). Existing methods customized for this task usually learn the generation from a large-scale dataset (Liu et al. 2016). Unlike them, our framework can also be applied to this task, but it generates the result using only one exemplar image. As shown in Fig. 8, our method can transfer the clothing or painting textures to other clothing (*e.g.*, (a) and (b)), virtually try on target clothing (*e.g.*, (c)), or edit the clothing shape (*e.g.*, (d)).

4.3 Comparisons

We compare our method with SOTA universal interactive texture transfer algorithms including two conventional methods (T-Effect (Yang et al. 2017) and CFITT (Men et al. 2018)), three neural-based methods (Neural Doodle (Champandard 2016), STROTSS (Kolkin, Salavon, and Shakhnarovich 2019), and Gatys2017 (Gatys et al. 2017)), and one GAN-based method (TuiGAN (Lin et al. 2020)).

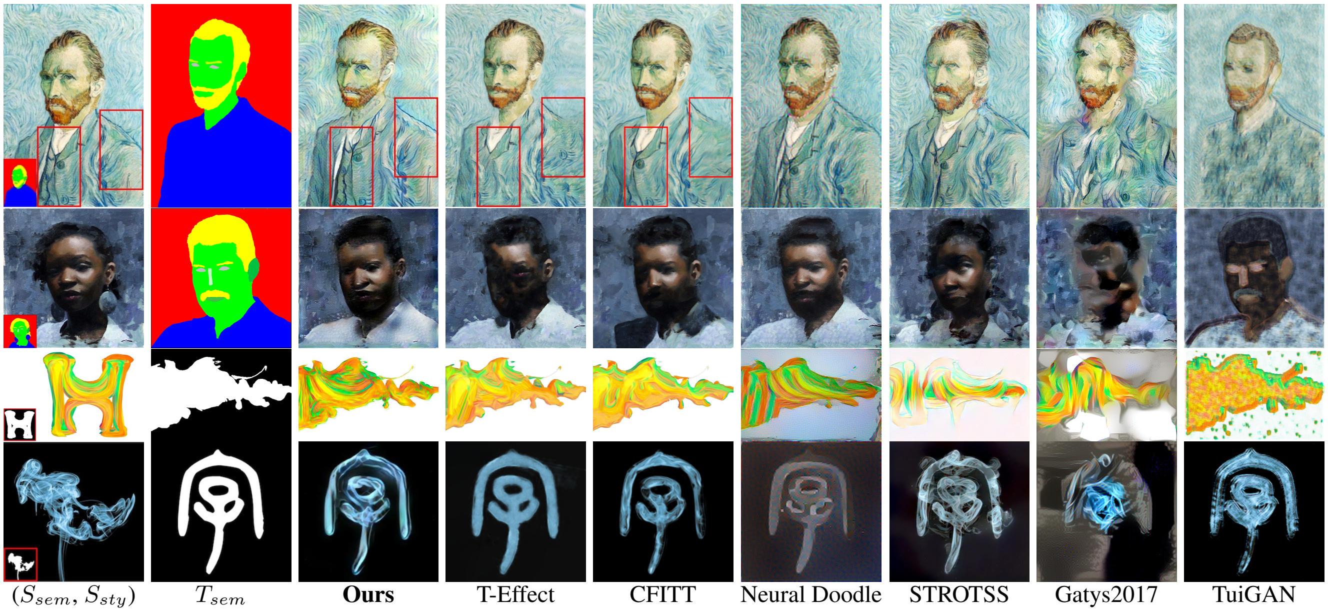


Figure 9: Qualitative comparison with the state-of-the-art universal interactive texture transfer methods. See more in *SM*.

Method[1]	256 × 256 (sec)		512 × 512 (sec)	
	CPU	GPU	CPU	GPU
T-Effect	101.52	-	241.85	-
CFITT	112.05	-	572.21	-
Neural Doodle	$\sim 2.8\times10^3$	178.78	$\sim 1.6\times10^4$	$\sim 1.1\times10^3$
STROTSS	$\sim 1.4\times10^3$	262.32	$\sim 3.9\times10^3$	668.84
Gatys2017	$\sim 2.2\times10^3$	144.78	$\sim 1.0\times10^4$	574.81
TuiGAN	-	$\sim 1.8\times10^4$	-	OOM
Ours	**2.573**	**0.232**	**12.381**	**0.956**

[1] Tested on a 3.3 GHz hexa-core CPU and a 6GB Nvidia 1060 GPU.

Table 1: Execution time comparison. OOM: out of memory.

For a fair comparison, we use their default settings except that the content weights of Neural Doodle, STROTSS, and Gatys2017 are set to 0, as there is no content image corresponding to T_{sem} in our task.

Qualitative Comparison. The qualitative results are shown in Fig. 9. Compared with T-Effect and CFITT, our method can synthesize higher quality results with better-preserved structures (*e.g.*, the red rectangle areas in the 1^{st} row, the face or clothing areas in the 2^{nd} row) and more vivid stylization effects (*e.g.*, the bottom two rows). For Neural Doodle, as it only bases on high-level features, it fails to reproduce clear images with low-level details and often introduces pixel noises. Moreover, STROTSS cannot achieve accurate semantic guidance, producing poor results with missing details (*e.g.*, the eyes in the 1^{st} row) and misaligned structures (*e.g.*, the 2^{nd} row). Gatys2017 matches the global statistics (*i.e.*, Gram matrix) for each semantic area, which cannot preserve the local texture structures. TuiGAN is hard to learn the underlying relationship between two images with a large domain gap (*e.g.*, semantic map and painting), thus cannot translate the exquisite texture details properly.

Efficiency. In Table 1, we compare the running time with the competitors. Compared with conventional methods T-Effect and CFITT on CPU, our method achieves 1-2 orders of magnitude faster in resolution 256 × 256 and 512 × 512. Our speed can be further accelerated by using a GPU card, eventually reaching 2-5 orders of magnitude faster than SOTA. Note that TuiGAN needs several hours and much more memory to train a model for each image pair.

User Study. We also conduct a user study to evaluate the quality quantitatively. Given unlimited time, 50 users are asked to select the favorite ones from 40 octets of images comprising three inputs (S_{sty}, S_{sem}, T_{sem}), and five randomly shuffled outputs (T-Effect, CFITT, Neural Doodle, STROTSS, and ours). We collect 2000 responses in total. The statistics indicate that our method achieves subjectively preferred results (**32.1%**) than T-Effect (23.8%), CFITT (26.2%), Neural Doodle (10.3%), and STROTSS (7.6%).

5 Conclusion

In this paper, we propose a novel neural-based framework, dubbed *texture reformer*, for fast and universal interactive texture transfer. A feed-forward multi-view and multi-stage synthesis procedure is imposed to synthesize high-quality results with coherent structures and fine texture details from coarse to fine. Moreover, we also introduce a novel learning-free *view-specific texture reformation (VSTR)* operation with a new semantic map guidance strategy to realize more accurate semantic-guided and structure-preserved texture transfer. Experimental results demonstrate the effectiveness of our framework on many texture transfer tasks. And compared with SOTA algorithms, it not only achieves higher quality results but also is 2-5 orders of magnitude faster.

Acknowledgements

This work was supported in part by the projects No. 2020YFC1523101, 19ZDA197, LY21F020005, 2021009, 2019C03137, NSFC project: research on key technologies of art image restoration based on decoupling learning (62172365), MOE Frontier Science Center for Brain Science & Brain-Machine Integration (Zhejiang University), and Key Scientific Research Base for Digital Conservation of Cave Temples (Zhejiang University), State Administration for Cultural Heritage.

References

An, J.; Huang, S.; Song, Y.; Dou, D.; Liu, W.; and Luo, J. 2021. ArtFlow: Unbiased Image Style Transfer via Reversible Neural Flows. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 862–871.

Barnes, C.; Shechtman, E.; Finkelstein, A.; and Goldman, D. B. 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (TOG)*, 28(3): 24.

Barnes, C.; Shechtman, E.; Goldman, D. B.; and Finkelstein, A. 2010. The generalized patchmatch correspondence algorithm. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 29–43. Springer.

Bookstein, F. L. 1989. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 11(6): 567–585.

Champandard, A. J. 2016. Semantic style transfer and turning two-bit doodles into fine artworks. *arXiv preprint arXiv:1603.01768*.

Chen, H.; Zhao, L.; Qiu, L.; Wang, Z.; Zhang, H.; Xing, W.; and Lu, D. 2020. Creative and diverse artwork generation using adversarial networks. *IET Computer Vision*, 14(8): 650–657.

Chen, H.; Zhao, L.; Wang, Z.; Ming, Z. H.; Zuo, Z.; Li, A.; Xing, W.; and Lu, D. 2021a. Artistic Style Transfer with Internal-external Learning and Contrastive Learning. In *Thirty-Fifth Conference on Neural Information Processing Systems (NeurIPS)*.

Chen, H.; Zhao, L.; Wang, Z.; Zhang, H.; Zuo, Z.; Li, A.; Xing, W.; and Lu, D. 2021b. DualAST: Dual Style-Learning Networks for Artistic Style Transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 872–881.

Chen, H.; Zhao, L.; Zhang, H.; Wang, Z.; Zuo, Z.; Li, A.; Xing, W.; and Lu, D. 2021c. Diverse image style transfer via invertible cross-space mapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 14880–14889.

Chen, T. Q.; and Schmidt, M. 2016. Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337*.

Chen, W.; and Hays, J. 2018. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 9416–9425.

Darabi, S.; Shechtman, E.; Barnes, C.; Goldman, D. B.; and Sen, P. 2012. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on graphics (TOG)*, 31(4): 1–10.

Efros, A. A.; and Freeman, W. T. 2001. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 341–346.

Efros, A. A.; and Leung, T. K. 1999. Texture synthesis by non-parametric sampling. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1033. IEEE.

Frigo, O.; Sabater, N.; Delon, J.; and Hellier, P. 2016. Split and match: Example-based adaptive patch sampling for unsupervised style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 553–561.

Frühstück, A.; Alhashim, I.; and Wonka, P. 2019. Tilegan: synthesis of large-scale non-homogeneous textures. *ACM Transactions on Graphics (TOG)*, 38(4): 1–11.

Gatys, L.; Ecker, A. S.; and Bethge, M. 2015. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 262–270.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2414–2423.

Gatys, L. A.; Ecker, A. S.; Bethge, M.; Hertzmann, A.; and Shechtman, E. 2017. Controlling perceptual factors in neural style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3985–3993.

Goferman, S.; Zelnik-Manor, L.; and Tal, A. 2011. Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(10): 1915–1926.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2672–2680.

Haeberli, P. 1990. Paint by numbers: Abstract image representations. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, 207–214.

Han, X.; Hu, X.; Huang, W.; and Scott, M. R. 2019a. Clothflow: A flow-based model for clothed person generation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 10471–10480.

Han, X.; Wu, Z.; Huang, W.; Scott, M. R.; and Davis, L. S. 2019b. Finet: Compatible and diverse fashion image inpainting. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 4481–4491.

Han, X.; Wu, Z.; Wu, Z.; Yu, R.; and Davis, L. S. 2018. Viton: An image-based virtual try-on network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7543–7552.

Hertzmann, A.; Jacobs, C. E.; Oliver, N.; Curless, B.; and Salesin, D. H. 2001. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 327–340.

Huang, X.; and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1501–1510.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1125–1134.

Jing, Y.; Liu, X.; Ding, Y.; Wang, X.; Ding, E.; Song, M.; and Wen, S. 2020. Dynamic instance normalization for arbitrary style transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, 4369–4376.

Jing, Y.; Liu, Y.; Yang, Y.; Feng, Z.; Yu, Y.; Tao, D.; and Song, M. 2018. Stroke controllable fast style transfer with adaptive receptive fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 238–254.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 694–711. Springer.

Kolkin, N.; Salavon, J.; and Shakhnarovich, G. 2019. Style Transfer by Relaxed Optimal Transport and Self-Similarity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 10051–10060.

Kwatra, V.; Essa, I.; Bobick, A.; and Kwatra, N. 2005. Texture optimization for example-based synthesis. *ACM Transactions on Graphics (TOG)*, 24(3): 795–802.

Li, C.; and Wand, M. 2016a. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2479–2486.

Li, C.; and Wand, M. 2016b. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 702–716. Springer.

Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; and Yang, M.-H. 2017. Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 386–396.

Liao, J.; Yao, Y.; Yuan, L.; Hua, G.; and Kang, S. B. 2017. Visual attribute transfer through deep image analogy. *ACM Transactions on Graphics (TOG)*.

Lin, J.; Pang, Y.; Xia, Y.; Chen, Z.; and Luo, J. 2020. TuiGAN: Learning Versatile Image-to-Image Translation with Two Unpaired Images. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 740–755. Springer.

Liu, Z.; Luo, P.; Qiu, S.; Wang, X.; and Tang, X. 2016. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1096–1104.

Lu, J.; Barnes, C.; DiVerdi, S.; and Finkelstein, A. 2013. RealBrush: painting with examples of physical media. *ACM Transactions on Graphics (TOG)*, 32(4): 1–12.

Lu, J.; Barnes, C.; Wan, C.; Asente, P.; Mech, R.; and Finkelstein, A. 2014. DecoBrush: drawing structured decorative patterns by example. *ACM Transactions on Graphics (TOG)*, 33(4): 1–9.

Lu, J.; Yu, F.; Finkelstein, A.; and DiVerdi, S. 2012. HelpingHand: Example-based stroke stylization. *ACM Transactions on Graphics (TOG)*, 31(4): 1–10.

Lu, M.; Zhao, H.; Yao, A.; Xu, F.; Chen, Y.; and Zhang, L. 2017. Decoder network over lightweight reconstructed feature for fast semantic style transfer. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2469–2477.

Men, Y.; Lian, Z.; Tang, Y.; and Xiao, J. 2018. A common framework for interactive texture transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6353–6362.

Mirza, M.; and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Myronenko, A.; and Song, X. 2010. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(12): 2262–2275.

Park, D. Y.; and Lee, K. H. 2019. Arbitrary style transfer with style-attentional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5880–5888.

Shaham, T. R.; Dekel, T.; and Michaeli, T. 2019. Singan: Learning a generative model from a single natural image. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 4570–4580.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Wang, Z.; Zhao, L.; Chen, H.; Qiu, L.; Mo, Q.; Lin, S.; Xing, W.; and Lu, D. 2020a. Diversified Arbitrary Style Transfer via Deep Feature Perturbation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7789–7798.

Wang, Z.; Zhao, L.; Chen, H.; Zuo, Z.; Li, A.; Xing, W.; and Lu, D. 2021. Evaluate and improve the quality of neural style transfer. *Computer Vision and Image Understanding (CVIU)*, 207: 103203.

Wang, Z.; Zhao, L.; Lin, S.; Mo, Q.; Zhang, H.; Xing, W.; and Lu, D. 2020b. GLStyleNet: exquisite style transfer combining global and local pyramid features. *IET Computer Vision*, 14(8): 575–586.

Yang, S.; Liu, J.; Lian, Z.; and Guo, Z. 2017. Awesome typography: Statistics-based text effects transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7464–7473.

Yang, S.; Wang, Z.; Wang, Z.; Xu, N.; Liu, J.; and Guo, Z. 2019. Controllable artistic text style transfer via shape-matching gan. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 4442–4451.

Zhang, C.; Zhu, Y.; and Zhu, S.-C. 2019. MetaStyle: Three-Way Trade-off among Speed, Flexibility, and Quality in Neural Style Transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, 1254–1261.

Zhao, L.; Mo, Q.; Lin, S.; Wang, Z.; Zuo, Z.; Chen, H.; Xing, W.; and Lu, D. 2020. UCTGAN: Diverse Image Inpainting Based on Unsupervised Cross-Space Translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5741–5750.

Zhu, J.-Y.; Krähenbühl, P.; Shechtman, E.; and Efros, A. A. 2016. Generative visual manipulation on the natural image manifold. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 597–613. Springer.