TE141K: Artistic Text Benchmark for Text Effect Transfer

Shuai Yang, *Student Member, IEEE*, Wenjing Wang, *Student Member, IEEE*, and Jiaying Liu, *Senior Member, IEEE*

Abstract—Text effects are combinations of visual elements such as outlines, colors and textures of text, which can dramatically improve its artistry. Although text effects are extensively utilized in the design industry, they are usually created by human experts due to their extreme complexity; this is laborious and not practical for normal users. In recent years, some efforts have been made toward automatic text effect transfer; however, the lack of data limits the capabilities of transfer models. To address this problem, we introduce a new text effects dataset, TE141K,[1] with 141,081 text effect/glyph pairs in total. Our dataset consists of 152 professionally designed text effects rendered on glyphs, including English letters, Chinese characters, and Arabic numerals. To the best of our knowledge, this is the largest dataset for text effect transfer to date. Based on this dataset, we propose a baseline approach called text effect transfer GAN (TET-GAN), which supports the transfer of all 152 styles in one model and can efficiently extend to new styles. Finally, we conduct a comprehensive comparison in which 14 style transfer models are benchmarked. Experimental results demonstrate the superiority of TET-GAN both qualitatively and quantitatively and indicate that our dataset is effective and challenging.

Index Terms—Text effects, style transfer, deep neural network, large-scale dataset, model benchmarking

1 Introduction

Text effects are additional style features for text, such as colors, outlines, shadows, stereoscopic effects, glows and textures. Rendering text in the style specified by the example text effects is referred to as text effect transfer. Applying visual effects to text is very common and important in graphic design. However, manually rendering text effects is labor intensive and requires great skill beyond the abilities of normal users. In this work, we introduce a large-scale text effect dataset to benchmark existing style transfer models on automatic text effect rendering and further propose a novel feature disentanglement neural network that can synthesize high-quality text effects on arbitrary glyphs.

Text effect transfer is a subtopic of general image style transfer. General image style transfer has been extensively studied in recent years. Based on style representation, it can be categorized into global and local methods. Global methods [1], [2], [3], [4], [5] represent styles as global statistics of image features and transfer styles by matching global feature distributions between the style image and the generated image. The most famous one is the pioneering neural style transfer [1], which exploits deep neural features and represents styles as Gram matrices [6]. However, the global representation of general styles does not apply to text effects. Text effects are highly structured along the glyph and cannot be simply characterized in terms of the mean, variance or other global statistics [2], [3], [4], [5] of the texture features. Instead, the text effects should be learned with the corresponding glyphs.

On the other hand, local methods [7], [8], [9] represent styles as local patches, and style transfer is essentially texture rearrangement, which seems to be more suitable for text effects than global statistics. In fact, the recent work of [9], which is the first study of text effect transfer, is a local method; textures are rearranged to correlated positions on text skeletons. However, it is difficult for local methods to preserve global style consistency. In addition, the patch-matching procedure of these methods usually suffers from a high computational complexity.

To handle a particular style, researchers have explored modeling styles from data rather than using general statistics or patches, which refers to image-to-image translation [10]. Early attempts [10], [11] trained generative adversarial networks (GANs) to map images from two domains; this technique is limited to only two styles. StarGAN [12] employs one-hot vectors to handle multiple predefined styles but requires expensive data collection and retraining to handle new styles. Despite these limitations, image-to-image translation methods have shown great success in generating vivid styles of building facades, street views, shoes, handbags, etc., from the corresponding datasets [13], [14], [15], [16]. However, the style of text effects is less well explored in this area due to the lack of related datasets.

An attempt to solve this problem uses MC-GAN [17], for which a small-scale dataset containing 910 images (35 text effects rendered on 26 capital letters) is collected. The authors [17] combined font transfer and text effect transfer

1. Project page: https://daooshee.github.io/TE141K/

• *The authors are with the Wangxuan Institute of Computer Technology, Peking University, Beijing 100871, China. E-mail: {williamyang, daooshee, liujiaying}@pku.edu.cn.*

Manuscript received 30 Apr. 2019; revised 12 Feb. 2020; accepted 19 Mar. 2020. Date of publication 1 Apr. 2020; date of current version 2 Sept. 2021.
(Corresponding author: Jiaying Liu.)
Recommended for acceptance by E. Shechtman.
Digital Object Identifier no. 10.1109/TPAMI.2020.2983697

Fig. 1. Representative text effects in TE141K. Text styles are grouped into three subsets based on glyph type, including TE141K-E (English alphabet subset, 67 styles), TE141K-C (Chinese character subset, 65 styles), and TE141K-S (symbol and other language subset, 20 styles).

using two successive subnetworks and trained them end-to-end using synthetic font data and the collected text effect dataset. The data-driven model shows good performance on this dataset. However, the limited size of the dataset can only support the model in handling capital letters with a small resolution of 64×64, which is far from meeting actual needs. Therefore, it is necessary to construct a large-scale dataset that has adequate style and glyph diversity for data-driven text effect transfer model design and benchmarking.

To address this practical issue, we develop TE141K, a large-scale dataset with 141,081 text effect/glyph pairs for data-driven text effect transfer, as shown in Fig. 1. TE141K contains 152 different kinds of professionally designed text effects collected from the Internet. Each style is rendered on a variety of glyphs, including English letters, Chinese characters, Japanese kana, and Arabic numerals, to form the style images. Besides these rendered in-the-wild styles, we design a simple yet effective style augmentation method [18] to obtain infinite synthetic styles, which can serve as a supplement to TE141K to improve the robustness of transfer models. Regarding content images, we further preprocess them so that they can provide more spatial information on glyphs, which makes it easier for the network to capture the spatial relationship between the glyph and the text effects.

Based on the large-scale dataset, we propose a novel approach for text effect transfer with two distinctive aspects. First, we develop a novel TET-GAN built upon encoder-decoder architectures. The encoders are trained to disentangle content and style features in the text effect images, while the decoders are trained to reconstruct features back to images. TET-GAN performs two functions, stylization and destylization, as shown in Fig. 2. Stylization is implemented by recombining the disentangled content and style features, while destylization is implemented solely by decoding content features. The task of destylization guides the network to precisely extract the content feature, which in turn helps the network better capture its spatial relationship with the style feature in the task of stylization. Through feature disentanglement, our network can simultaneously support hundreds of distinct styles, whereas traditional image-to-image translation methods [10] only deal with two styles. Second, we propose a self-stylization training scheme for one-reference style transfer. Leveraging the knowledge learned from our dataset, the network only needs to be finetuned on one reference example, and then it can render the new user-specified style on any glyph, providing much more flexibility than StarGAN [12].

Compared with our previous work [18], we expand TET-GAN to joint font style and text effect transfer, which achieves better style consistency between the output and the reference style. We further explore semisupervised text effect transfer to improve the model's generalization. In addition, we present analyses of the features extracted by TET-GAN to validate the disentanglement of styles and content, which helps clarify the working mechanism of the model. Finally, in addition to our enclosed conference paper, we focus on the construction and analysis of the new large-scale dataset TE141K and conduct more comprehensive and in-depth experiments for model benchmarking, including 1) a new dataset 160 percent larger than the old one [18] in terms of styles and glyphs, 2) objective and subjective quantitative evaluations over fourteen style transfer models on three text effect transfer tasks, and 3) analyses on the performance-influencing factors of text effect transfer. In summary, our contributions are threefold:

- We introduce a large dataset named TE141K containing thousands of professionally designed text effect images, which we believe can be useful for the research areas of text effect transfer, multidomain transfer, image-to-image translation, etc.
- We propose a novel TET-GAN to disentangle and recombine glyph features and style features for text effect transfer. The explicit content and style representations enable effective stylization and destylization on multiple text effects. A novel self-stylization training scheme for style extension is further proposed to improve the flexibility of the network.
- We provide a comprehensive benchmark for our method and state-of-the-art methods, which validates the challenges of the proposed dataset and the superiority of our feature disentanglement model.

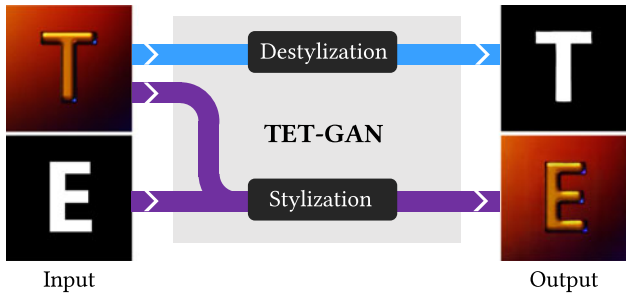


Fig. 2. Our TET-GAN implements two functions: destylization for removing style features from the text and stylization for transferring the visual effects from highly stylized text onto other glyphs.

TABLE 1
A Summary of TE141K

	# Style	# Glyphs	Glyph Types	# Training/# Testing	# Images
TE141K-E	67	988	52 English Alphabets in 19 Fonts	874/114	59,280
TE141K-C	65	837	775 Chinese Characters, 52 English Alphabets, 10 Arabic Numerals	740/97	54,405
TE141K-S	20	1,024	56 Special Symbols, 968 Letters in Japanese, Russian, etc.	900/124	20,480
Total	152	2,849		2,514/335	141,081

Based on glyph types, TE141K can be split into three subsets, where styles are different in different subsets.

The rest of this paper is organized as follows. Section 2 presents our new dataset. In Section 3, we review representative style transfer models for benchmarking. In Section 4, the details of the proposed TET-GAN for text effect transfer are presented. Section 5 benchmarks the proposed method and the state-of-the-art style transfer models. Finally, we conclude our work in Section 6.

2 A Large-Scale Dataset for Text Effect Transfer

In this section, we will introduce the details of TE141K and analyze its data distribution.

2.1 Data Collection

We built a dataset with the help of automation tools in Adobe Photoshop. Specifically, we first collected PSD files of text effects released by several text effect websites and PSD files we created by following tutorials on these websites. Then, we used batch tools and scripts to automatically replace the glyphs and produce approximately one thousand text effect images for each PSD file. There are also two text effects kindly provided by Yang *et al.* [9], adding up to 152 different kinds of text styles. Finally, we obtained 141,081 text effect images with a resolution of 320×320 and their corresponding glyph images to form TE141K. Based on glyph types, we divide TE141K into three subsets, where text styles are different in different subsets. Fig. 1 and Table 1 show an overview of these three subsets, including:

- *TE141K-E.* This subset contains 67 styles (59,280 image pairs, 988 glyphs per style), where all glyphs are English alphabets, making text effects easier to transfer compared to the other two subsets. This subset serves as a baseline to explore multistyle transfer.
- *TE141K-C.* This subset contains 65 styles (54,405 image pairs, 837 glyphs per style). The glyphs for training are all Chinese characters, while the glyphs for testing contain Chinese characters, English alphabets and Arabic numerals. This subset can be used to test the glyph generalization ability of the transfer model.
- *TE141K-S.* This subset contains 20 styles (20,480 image pairs, 1,024 glyphs per style). The glyphs are special symbols and letters of common languages other than Chinese and English. In this paper, we use this subset for one-reference training to test the flexibility (efficiency of new style extension) of the transfer model.

For each subset and each kind of text effect, we use approximately 87 percent of the images for training and 13 percent for testing.

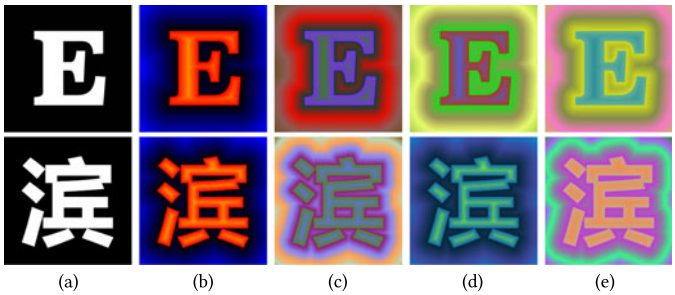


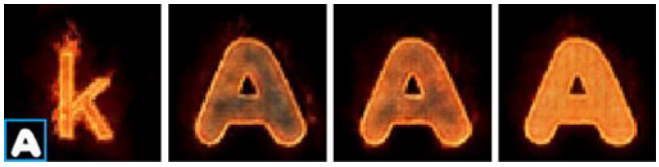
Fig. 3. Distribution-aware data augmentation. (a) Raw text image. (b) Results of distribution-aware text image preprocessing (the contrast is enhanced for better visualization). (c)-(e) Results of distribution-aware text effect augmentation by tinting (b) using random colormaps.

2.2 Data Processing

Distribution-Aware Text Image Preprocessing. As reported in [9], the spatial distribution of the texture in text effects is highly related to its distance from the glyph, forming an effective prior for text effect transfer. To leverage this prior, we propose a distribution-aware text image preprocessing to directly feed models trained on TE141K with distance cues. As shown in Fig. 3, we extend the raw text image from one channel to three channels. The R channel is the original text image, while the G channel and B channel are distance maps where the value of each pixel is its distance to the background black region and the foreground white glyph, respectively. Another advantage of the preprocessing is that our three-channel text images have much fewer saturated areas than the original ones, which greatly facilitates the extraction of valid features.

Distribution-Aware Text Effect Augmentation. In addition to the text images, we further propose the distribution-aware augmentation of the text effect images. The key idea is to augment our training data by generating random text effects based on the pixel distance from the glyph. Specifically, we first establish a random colormap for each of the R and G channels, which maps each distance value to a corresponding color. Then, we use the colormaps of the R and G channels to tint the background black region and the foreground white glyph in the text image separately. Figs. 3c, 3d, and 3e show examples of the randomly generated text effect images. With colors that reflect structural distribution, these images can effectively guide transfer models to identify the spatial relationship between the text effects and the glyphs. In addition, data augmentation could also increase the generalization capabilities of the model.

In Fig. 4, we examine the effect of our distribution-aware text image preprocessing and text effect augmentation on TE141K through a comparative experiment with the model



(a) input (b) w/o pre+aug (c) w/o aug (d) pre+aug

Fig. 4. A comparison of results with and without our distribution-aware data preprocessing and augmentation.

proposed in Section 4. Without preprocessing and augmentation, the inner flame textures are not synthesized correctly. As seen in Fig. 4d, our distribution-aware data augmentation strategy helps the network learn to infer textures based on their correlated position on the glyph and synthesize better flame textures.

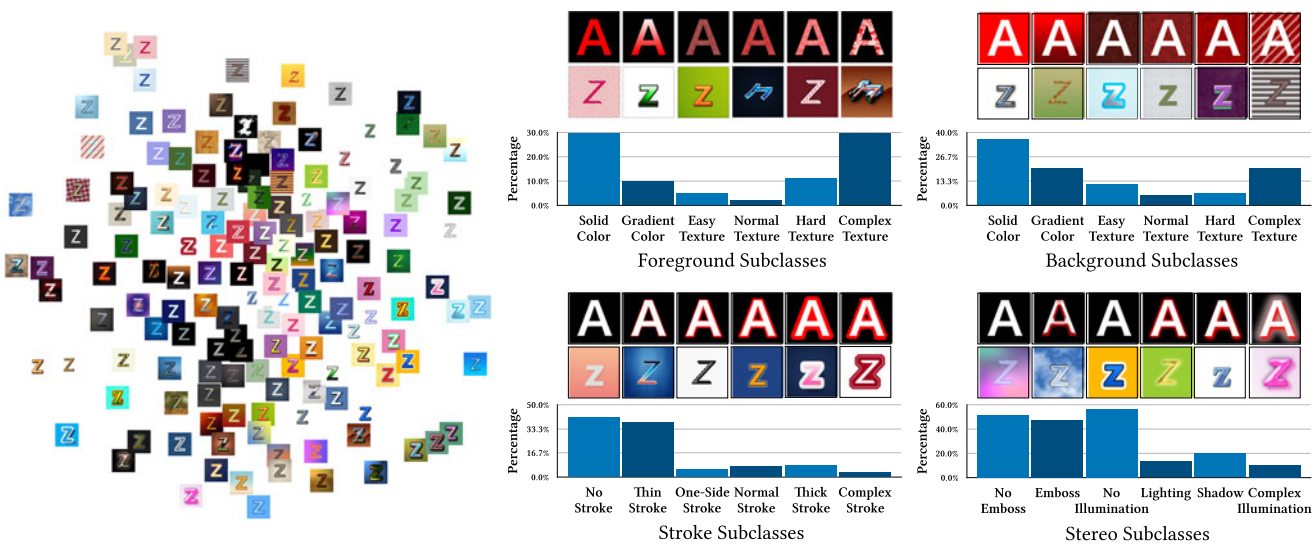
2.3 Dataset Statistics

In the proposed dataset, there are a wide variety of text effects. To explore their distribution, we first obtain their feature maps from the fifth layer of the VGG network [20]. Then, we conduct nonlinear dimensionality reduction by t-SNE [19] and visualize the distribution of text effects. We choose VGG features instead of RGB information because text effects with similar structural characteristics are close to each other in VGG features. As illustrated in Fig. 5a, text effects are evenly distributed as a circle, indicating the diversity and richness of the text effects in our dataset.

Text effects usually consist of multiple fundamental visual elements, such as textures, stroke outlines and stereo effects. These factors may vary greatly and make text effect transfer a challenging task. To better investigate how they affect the model performance, we quantify these factors and manually label the text effects with the corresponding factors. We will introduce the statistics of these factors in this section and investigate their influence on the model performance in Section 5.6.

We consider texture in two categories: foreground and background. For background texture, based on complexity level, we classify it into 6 subclasses: *Solid Color*, *Gradient Color*, *Easy Texture*, *Normal Texture*, *Hard Texture* and *Complex Texture*. *Solid Color* is the simplest; transfer models only need to directly copy the original color. *Gradient Color* is more complex; the distribution of color is directional. For the other four subclasses, there are textures on the background. In *Easy Texture*, textures are imperceptible to human eyes. From *Normal Texture* to *Hard Texture* and *Complex Texture*, textures become distinct and irregular. As shown in Fig. 5b, 36 percent of the background is solid, and 20 percent is of gradient color, which is consistent with the fact that background is often set to be simple to better emphasize the foreground glyph. Similarly, we classify foreground into the same 6 subclasses. In contrast to the background, 30 percent of the foreground has *Complex Texture*. This may be because the foreground is the main body of text effects; therefore, it is often fully decorated to increase artistic quality.

Stroke outlines and stereo effects are two common elements used in cartoon and 3D text effects, respectively. Based on complexity and thickness, we classify strokes into 6 subclasses: *No Stroke*, *Thin Stroke*, *One-Side Stroke*, *Normal Stroke*, *Thick Stroke* and *Complex Stroke*. We observe that the thickness of many *Thin Stroke* and *Normal Stroke* effects is uneven in direction; therefore, we further divide them into a more specific class, *One-Side Stroke*. In TE141K, 59 percent of the text effects contain strokes. Stereo effects are usually a combination of multiple special effects, such as emboss and illumination effects. In addition, illumination effects can be further classified into lighting and shadow. Similarly, we classify stereo effects into 6 subclasses: *No Emboss*, *Emboss*, *No Illumination*, *Lighting*, *Shadow*, and *Complex Illumination*, where *Complex Illumination* is a combination of more than two illumination effects. In TE141K, 48 percent of the text effects contain embossing, and 43 percent contain illumination effects.



(a) Visualization of VGG features of all text effects. (b) Distribution of background, foreground, stroke, and stereo effects subclasses.

Fig. 5. Statistics of TE141K. (a) Visualized text effect distribution in TE141K by t-SNE [19] based on their VGG features [20]. Perceptually similar text effects are placed close to each other. The even distribution indicates the diversity and richness of our dataset. (b) Distribution of different background, foreground, stroke, and stereo effects subclasses of our dataset. Representative images are shown at the top. The first row contains schematics where red is used to represent specific text effect subclasses. The second row contains representative samples from TE141K.

TABLE 2
Summary of Benchmarking Representative Style Transfer Models and the Proposed TET-GAN, Showing the Model Type, Model Names, Number of Style Supported per Model (# Style), Support for Supervised One-Reference (SOR) or Unsupervised One-Reference (UOR) Style Transfer, Availability of Feed-Forward Fast Style Transfer, Target Type (General Image Style or Text Effect Transfer), Usage of Deep Models and the Style Representation

Type	Model	Flexibility		Efficiency	Model Design		
		# Style	SOR/UOR	Feed-Forward	Type	Deep	Style Representation
Global	NST [1]	∞	UOR	$\times$	general	✓	Gram matrix of deep features
	AdaIN [3]	∞	UOR	✓	general	✓	mean and variance of deep features
	WCT [5]	∞	UOR	✓	general	✓	mean and covariance of deep features
Local	Analogy [21]	∞	SOR	$\times$	general	$\times$	image patches
	Quilting [22]	∞	UOR	$\times$	general	$\times$	image patches
	CNNMRF [7]	∞	UOR	$\times$	general	✓	feature patches
	Doodles [23]	∞	SOR	$\times$	general	✓	feature patches
	T-Effect [9]	∞	SOR	$\times$	text	$\times$	image patches
	UT-Effect [24]	∞	UOR	$\times$	text	$\times$	image patches
GAN-based	Pix2pix [10]	1/N*	$\times$	✓	general	✓	learned features
	BicycleGAN [25]	1/N*	$\times$	✓	general	✓	learned features
	StarGAN [12]	N	$\times$	✓	general	✓	learned features
	MC-GAN [17]	∞	$\times$	✓	text	✓	learned features
	TET-GAN (ours)	N	$\times$	✓	text	✓	disentangled content and style features
	TET-GAN+ (ours)	∞	SOR/UOR	✓	text	✓	disentangled content and style features

** Pix2pix and BicycleGAN were originally designed to support only one style per model. In our experiment, we add an extra conditional text-style pair to their original input, which enables them to handle multiple styles in one model. Note: Compared to TET-GAN, TET-GAN+ additionally makes use of the proposed one-reference finetuning strategy for style extension.*

2.4 Comparison With Existing Datasets

Datasets play an important role in the development of neural networks. To the best of our knowledge, the dataset provided in the work on MC-GAN [17] is the only text effect dataset in the literature. To train MC-GAN, the authors collected 35 different kinds of text effects from the Internet. For each style, only an extremely limited set of 26 images of capital letters with a small size of 64×64 are rendered, forming a total of 910 style images, which cannot support training a network that is robust enough to produce high-resolution images of arbitrary glyphs. Therefore, MC-GAN can only handle 26 capital letters with a low resolution. In contrast, our TE141K contains 152 different kinds of text effects. For each style, at least 837 glyphs are rendered, adding up to 141,081 image pairs in total. In addition, the image size reaches 320×320. The proposed dataset exceeds that of [17] in terms of both quantity and diversity, supporting transfer models to render exquisite text effects on various glyphs.

2.5 Text Effect Transfer Tasks

On TE141K, we design three text effects transfer tasks according to the amount of information provided to transfer models, which will be benchmarked in Section 5:

- *General Text Effect Transfer*. In this task, text styles in the training and testing phases are the same. Benchmarking is conducted with all three dataset subsets. During testing, the models are provided with an input example text effect image, its glyph counterpart and the target glyph. This task is relatively easy, since models can become familiar with text effects through a large amount of training data. The challenge lies in transferring multiple styles in one model and generalizing to unseen glyphs.
- *Supervised One-Reference Text Effect Transfer*. In this task, text effects in the training and testing phases are different. For data-driven models, only the training sets of TE141K-E and TE141K-C are provided, and benchmarking is conducted on the testing set of TE141K-S. This task is more difficult, since models have to learn new text effects with only one example pair.
- *Unsupervised One-Reference Text Effect Transfer*. This task is similar to supervised one-reference text effect transfer except that during testing, the glyph image of the example text effect image is not provided. This task is the most difficult, since transfer models have to distinguish the foreground and background by themselves.

3 Benchmarking Existing Style Transfer Models

In this section, we briefly introduce existing representative style transfer models, which will be benchmarked on the proposed TE141K dataset in Section 5. These models can be categorized based on their style representations, i.e., global statistics, local patches and learned features. The choice of style representation can largely affect the characteristics of the model in terms of flexibility and efficiency. To give an intuitive comparison, we summarize the models and their characteristics in Table 2.

Global Models. Global models represent image styles as global statistics of image features and transfer styles by matching the global feature distributions between the style image and the generated image. The advantage of explicitly defined style representation is that any input style can be modelled and transferred without requiring a large paired dataset, which is suitable for the task of unsupervised one-reference transfer.

- *Neural Style Transfer (NST)*: The trend of parametric deep-based style transfer began with the pioneering work of neural style transfer [1]. In NST, Gatys *et al.* formulated image styles as the covariance of deep features in the form of a Gram matrix [6] and transferred style by matching high-level representations of the content image and the Gram matrices; this technique demonstrates the remarkable representative power of convolutional neural networks (CNN) to model style. The main drawback of NST is its computationally expensive optimization procedure. Follow-up work [26], [27], [28], [29] has been proposed to speed up NST by training a feed-forward network to minimize the loss of NST. However, efficiency is achieved at the expense of flexibility and quality. Thus, we select NST as our benchmarking global model.
- *Arbitrary Style Transfer (AdaIN)*: AdaIN [3] presents a feature transformation framework, where the style is represented by the mean and variance of deep features. By aligning the statistics of the content features with those of the style features via adaptive instance normalization (AdaIN), AdaIN allows for fast arbitrary style transfer, achieving flexibility and efficiency simultaneously.
- *Universal Style Transfer (WCT)*: WCT [5] follows the feature transformation framework and represents style as the covariance of deep features, which can be adjusted by whitening/coloring transforms (WCT). Compared to the variance in AdaIN [3], covariance can better capture high-level representations of the style.

Local Models. Local models represent styles as local patches and transfer styles by rearranging style patches to fit the structure of the content image. Similar to global models, local models are capable of extracting style from only a single style image, and are therefore suitable for the task of one-reference transfer. Compared to global models, local patches better depict style details. However, matching patches can be time-consuming.

- *Image Analogy (Analogy)*: Hertzmann *et al.* first presents a supervised framework called image analogy [21], which aims to learn the transformation between an unstylized source image and the corresponding stylized image. Style transfer is realized by applying the learned transformation to the target image. In [21], the transformation is realized by replacing unstylized image patches with the corresponding stylized ones.
- *Image Quilting (Quilting)*: Image quilting [22] rearranges image patches from the style image according to the intensity or gradient of the content image, which can transfer style in an unsupervised manner.
- *CNNMRF*: CNNMRF [7] combines CNN with an MRF regularizer and models image style by local patches of deep features; it is suitable for fine structure preservation and semantic matching. However, it fails when the content and style images have strong semantic differences due to patch mismatches.
- *Neural Doodles (Doodle)*: Neural doodles [23] introduce a supervised version of CNNMRF [7] by incorporating the semantic map of the style image as guidance, which alleviates the problem of patch mismatches.
- *Text Effect Transfer (T-Effect)*: Yang *et al.* proposed the first text effect transfer method, named T-Effect [9]. The authors modelled the text style by both the appearance of the patches and their correlated positions on the glyph, which achieved spatial consistency in text style.
- *Unsupervised Text Effect Transfer (UT-Effect)*: UT-Effect [24] is an unsupervised version of T-Effect [9], which generates new text effects from texture images rather than given text effects. It further exploits the saliency of the textures to enhance the legibility of the rendered text.

GAN-Based Models. GAN-based models are tasked to map between two or more domains; during this process, the style representation is implicitly learned from the data. This task-specific style representation has the advantage of producing vivid results but needs a large dataset for training and is usually hard to extend to new styles.

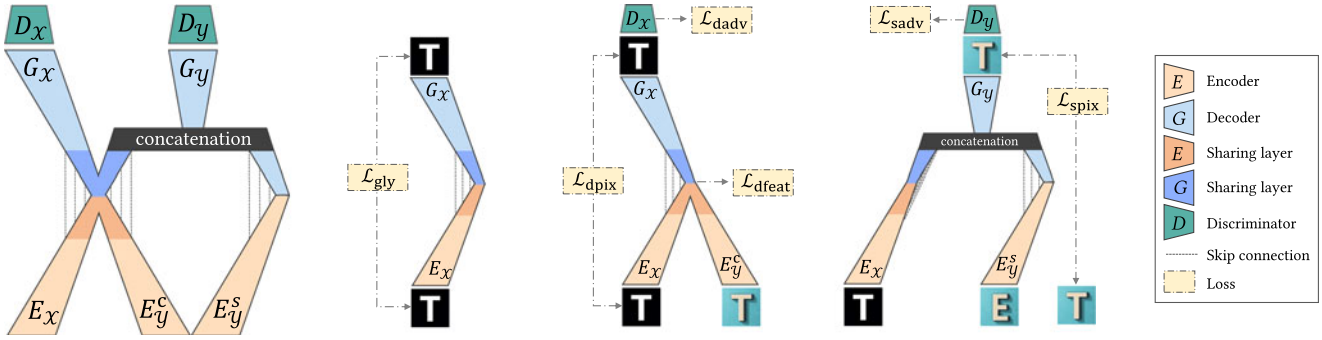
- *Pix2pix-cGAN (Pix2pix)*: Pix2pix [10] presents a general-purpose framework built upon U-Net [30] and PatchGAN [10] for the task of image-to-image translation and has shown high performance in many applications, such as style transfer, colorization, semantic segmentation and daytime hallucination. Despite its high performance, Pix2pix [10] is designed for two domains and thus has limited flexibility in handling multiple styles.
- *BicycleGAN*: BicycleGAN [25] tackles the problem of generating diverse outputs by encouraging bijective consistency between the output and learned latent features. However, it is ineffective in multistyle transfer.
- *StarGAN*: StarGAN [12] utilizes additional one-hot vectors as input to specify the target domain so that the network can learn a mapping between multiple domains, which is more flexible.
- *Multi-Content GAN (MC-GAN)*: Azadi *et al.* [17] presented an MC-GAN for the stylization of capital letters, which combines font transfer and text effect transfer using two successive subnetworks. A leave-one-out approach is introduced for few-reference style transfer (it takes about 6 example images as input, as reported in [17]), making the model more flexible. However, MC-GAN is designed to handle only 26 capital letters with a small image resolution of 64×64, which highly limits its uses.

4 TET-GAN for Text Effect Transfer

To construct a baseline model for TE141K, we propose a deep-based approach named TET-GAN to disentangle and recombine the content and style features of text effect images so that it can simultaneously handle multiple styles in TE141K. We further propose a one-reference finetuning strategy that flexibly extends TET-GAN to new styles.

4.1 Network Architecture and Loss Function

Our goal is to learn a two-way mapping between two domains $\mathcal{X}$ and $\mathcal{Y}$, which represent a collection of text images and text effect images, respectively. As shown in Fig. 6, TET-GAN consists of two content encoders $\{E_{\mathcal{X}}, E_{\mathcal{Y}}^{c}\}$, a style



(a) Network Architecture (b) Autoencoder (c) Destylization (d) Stylization

Fig. 6. The TET-GAN architecture. (a) An overview of the TET-GAN architecture. Our network is trained via three objectives: an autoencoder, destylization and stylization. (b) A glyph autoencoder to learn content features. (c) Destylization by disentangling content features from text effect images. (d) Stylization by combining content and style features.

encoder $\{E^s_{\mathcal{Y}}\}$, two domain generators $\{G_{\mathcal{X}}, G_{\mathcal{Y}}\}$ and two domain discriminators $\{D_{\mathcal{X}}, D_{\mathcal{Y}}\}$. $E_{\mathcal{X}}$ and $E^c_{\mathcal{Y}}$ map text images and text effect images, respectively, onto a shared content feature space, while $E^s_{\mathcal{Y}}$ maps text effect images onto a style feature space. $G_{\mathcal{X}}$ generates text images from the encoded content features. $G_{\mathcal{Y}}$ generates text effect images conditioned on both the encoded content and style features. Based on the assumption that domains $\mathcal{X}$ and $\mathcal{Y}$ share a common content feature space, we share the weights between the last few layers of $E_{\mathcal{X}}$ and $E^c_{\mathcal{Y}}$ as well as the first few layers of $G_{\mathcal{X}}$ and $G_{\mathcal{Y}}$. Discriminators are trained to distinguish the generated images from the real ones.

TET-GAN is trained on three tasks: *Glyph Autoencoder* $G_{\mathcal{X}} \circ E_{\mathcal{X}} : \mathcal{X} \to \mathcal{X}$ for learning content feature encoding, *Destylization* $G_{\mathcal{X}} \circ E^c_{\mathcal{Y}} : \mathcal{Y} \to \mathcal{X}$ for learning content feature disentanglement from text effect images, and *Stylization* $G_{\mathcal{Y}} \circ (E_{\mathcal{X}}, E^s_{\mathcal{Y}}) : \mathcal{X} \times \mathcal{Y} \to \mathcal{Y}$ for learning style feature disentanglement and combination with content features. Therefore, our objective is to solve the min-max problem

$$\min_{E,G} \max_{D} \mathcal{L}_{\text{gly}} + \mathcal{L}_{\text{desty}} + \mathcal{L}_{\text{sty}}, \tag{1}$$

where $\mathcal{L}_{\text{gly}}$, $\mathcal{L}_{\text{desty}}$ and $\mathcal{L}_{\text{sty}}$ are losses related to the glyph autoencoder, destylization, and stylization, respectively.

Glyph Autoencoder. First, the encoded content feature is required to preserve the core information of the glyph. Thus, we impose an autoencoder L_1 loss to force the content feature to completely reconstruct the input text image

$$\mathcal{L}_{\text{gly}} = \lambda_{\text{gly}} \mathbb{E}_x[\|G_{\mathcal{X}}(E_{\mathcal{X}}(x)) - x\|_1]. \tag{2}$$

Destylization. For destylization, we sample from the training set a text-style pair (x, y). We would like to map x and y onto a shared content feature space, where the feature can be used to reconstruct x, leading to the L_1 loss

$$\mathcal{L}_{\text{dpix}} = \mathbb{E}_{x,y}[\|G_{\mathcal{X}}(E^c_{\mathcal{Y}}(y)) - x\|_1]. \tag{3}$$

Furthermore, we would like $E^c_{\mathcal{Y}}$ to approach the ideal content feature extracted from x. To enforce this constraint, we formulate the feature loss as

$$\mathcal{L}_{\text{dfeat}} = \mathbb{E}_{x,y}[\|S_{\mathcal{X}}(E^c_{\mathcal{Y}}(y)) - z\|_1], \tag{4}$$

where $S_{\mathcal{X}}$ represents the sharing layers of $G_{\mathcal{X}}$ and $z = S_{\mathcal{X}}(E_{\mathcal{X}}(x))$. Our feature loss guides the content encoder $E^c_{\mathcal{Y}}$ to remove the style elements from the text effect image, preserving only the core information of the glyph.

Finally, we impose conditional adversarial loss to improve the quality of the generated results. $D_{\mathcal{X}}$ learns to determine the authenticity of the input text image and whether it matches the given text effect image. At the same time, $G_{\mathcal{X}}$ and $E^c_{\mathcal{Y}}$ learn to confuse $D_{\mathcal{X}}$

$$\begin{aligned}\mathcal{L}_{\text{dadv}} = \mathbb{E}_{x,y}[\log D_{\mathcal{X}}(x, y)] \\ - \mathbb{E}_y[\log(1 - D_{\mathcal{X}}(G_{\mathcal{X}}(E^c_{\mathcal{Y}}(y)), y))].\end{aligned} \tag{5}$$

The total loss for destylization takes the following form:

$$\mathcal{L}_{\text{desty}} = \lambda_{\text{dpix}}\mathcal{L}_{\text{dpix}} + \lambda_{\text{dfeat}}\mathcal{L}_{\text{dfeat}} + \lambda_{\text{dadv}}\mathcal{L}_{\text{dadv}}. \tag{6}$$

Stylization. For the task of stylization, we sample from the training set a text-style pair (x, y) and a text effect image y' that shares the same style with y but has a different glyph. We first extract the content feature from x and the style feature from y', which are then concatenated and fed into $G_{\mathcal{Y}}$ to generate a text effect image to approximate the ground-truth y in an L_1 sense

$$\mathcal{L}_{\text{spix}} = \mathbb{E}_{x,y,y'}[\|G_{\mathcal{Y}}(E_{\mathcal{X}}(x), E^s_{\mathcal{Y}}(y')) - y\|_1], \tag{7}$$

and confuse $D_{\mathcal{Y}}$ with conditional adversarial loss

$$\begin{aligned}\mathcal{L}_{\text{sadv}} = \mathbb{E}_{x,y,y'}[\log D_{\mathcal{Y}}(x, y, y')] \\ - \mathbb{E}_{x,y'}[\log(1 - D_{\mathcal{Y}}(x, G_{\mathcal{Y}}(E_{\mathcal{X}}(x), E^s_{\mathcal{Y}}(y')), y'))].\end{aligned} \tag{8}$$

Our final loss for stylization is

$$\mathcal{L}_{\text{sty}} = \lambda_{\text{spix}}\mathcal{L}_{\text{spix}} + \lambda_{\text{sadv}}\mathcal{L}_{\text{sadv}}. \tag{9}$$

4.2 One-Reference Text Effect Transfer

As introduced in Section 3, GAN-based methods are by nature heavily dependent on datasets and usually require thousands of training images, which greatly limits their applicability. To build a baseline that supports personalized style transfer, as the global and local methods do, we propose an one-reference finetuning strategy for style extension, where only one example style image is required.

One-Reference Supervised Learning. For an unseen style in Fig. 7a, as shown in the top row of Fig. 7c, TET-GAN trained on TE141K-C fails to synthesize texture details. To solve this

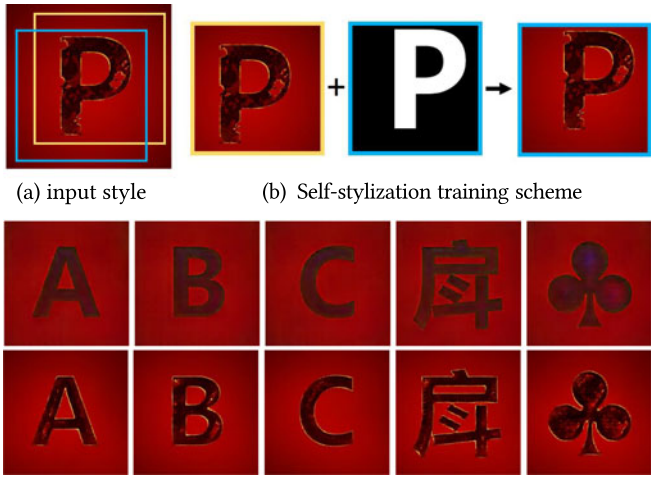


Fig. 7. One-reference text effects transfer. (a) New, user-specified text effects. (b) Random crop of the style image to generate image pairs for training. (c) Top row: Stylization result on an unseen style. Bottom row: Stylization result after one-reference finetuning. The model finetuned over (a) is able to transfer text effects onto other unseen characters.

problem, inspired by self-supervised adversarial training [31], we propose a simple yet efficient "self-stylization" training scheme. As shown in Fig. 7b, we randomly crop the images to obtain many text effect patches that have the same style but differ in the pixel domain. In other words, x, y, and y' in Eqs. (1)-(9) are patches cropped from the given image pair. They constitute a training set to finetune TET-GAN so that it can learn to reconstruct vivid textures, as shown in the bottom row of Fig. 7c. Note that our model finetuned over a single image can generalize well to other very different glyphs. Compared to the texture synthesis task [31], our one-reference style transfer task is more challenging, which requires generalization to glyphs. Thus, our training scheme further requires the model to be pretrained on a large amount of supervised data to learn the domain knowledge of the glyphs. As we will show later in Fig. 14, such domain knowledge learnt from TE141K plays an important role in improving the one-reference text effect transfer.

One-Reference Unsupervised Learning. For an unseen style y without a provided text image x, it is intuitive to exploit our destylization submodule to generate x from y and transform this one-reference unsupervised problem to a supervised one. In other words, $\tilde{x} = G_{\mathcal{X}}(E^c_{\mathcal{Y}}(y))$ is used as an auxiliary x during finetuning. Considering that the accuracy of the content features extracted from $\tilde{x}$ cannot be guaranteed, a style reconstruction loss is employed to further constrain the content features to help reconstruct y

$$\mathcal{L}_{\text{srec}} = \lambda_{\text{srec}} \mathbb{E}_y[\|G_{\mathcal{Y}}(E^c_{\mathcal{Y}}(y), E^s_{\mathcal{Y}}(y)) - y\|_1]. \tag{10}$$

Our objective for unsupervised learning takes the form

$$\min_{E,G} \max_{D} \mathcal{L}_{\text{gly}} + \mathcal{L}_{\text{desty}} + \mathcal{L}_{\text{sty}} + \mathcal{L}_{\text{srec}}. \tag{11}$$

The model of TET-GAN combining the one-reference finetuning strategy is named TET-GAN+.

4.3 Semi-Supervised Text Effects Transfer

With the help of paired data provided by the proposed TE141K, we can explore the potential of TET-GAN in semisupervised learning, where the model is givensufficient but unpaired data for style extension. The main challenge is to establish an effective mapping between the style data and the glyph data. Inspired by the adversarial augmentation proposed by [32], we propose a hybrid supervised and unsupervised learning framework of TET-GAN, where two augmentation discriminators $D^{\text{aug}}_{\mathcal{X}}$ and $D^{\text{aug}}_{\mathcal{Y}}$ are introduced to receive unlabeled data. Specifically, $D^{\text{aug}}_{\mathcal{X}}$ is tasked to only judge the authenticity of the input text image without the need to determine whether it matches the given text effect image, and it uses the following objective function:

$$\begin{aligned}\mathcal{L}_{\text{daug}} = {} & \mathbb{E}_x[\log D^{\text{aug}}_{\mathcal{X}}(x)] \\ & - \mathbb{E}_y[\log(1 - D^{\text{aug}}_{\mathcal{X}}(G_{\mathcal{X}}(E^c_{\mathcal{Y}}(y))))].\end{aligned} \tag{12}$$

The objective function of $D^{\text{aug}}_{\mathcal{Y}}$ is similarly defined

$$\begin{aligned}\mathcal{L}_{\text{saug}} = {} & \mathbb{E}_{y,y'}[\log D^{\text{aug}}_{\mathcal{Y}}(y, y')] \\ & - \mathbb{E}_{x,y'}[\log(1 - D^{\text{aug}}_{\mathcal{Y}}(G_{\mathcal{Y}}(E_{\mathcal{X}}(x), E^s_{\mathcal{Y}}(y')), y'))].\end{aligned} \tag{13}$$

The advantage is that the supervised data serve as an anchor to force the model to generate outputs consistent with the inputs, while the unsupervised data can teach the model to deal with a wider variety of glyph and style inputs.

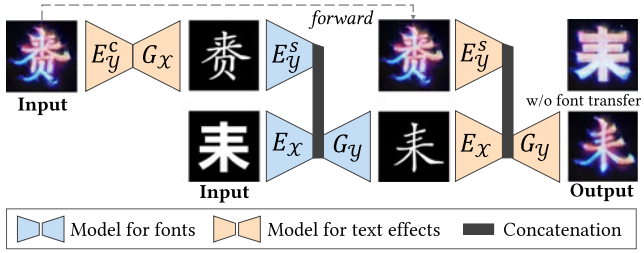


Fig. 8. Hybrid font style and text effect transfer framework. Two TET-GANs trained on the font dataset and text effects dataset constitute a uniform framework to transfer both font style and text effects.

4.4 Joint Font Style and Text Effect Transfer

As we will show later, TET-GAN has a good generalizability across fonts and is capable of transferring the text effects on a reference image to other glyphs in different font styles. However, some text effects are designed specifically for certain fonts. In Fig. 8, we show a case where the neon style suits a serif font (regular script) but not a sans-serif font (Microsoft Yahei). Thus, it is a good option to match the font styles. In this section, we explore the potential for TET-GAN in transferring the font style itself. Specifically, we choose an anchor font $\mathcal{F}_0$ (in this paper, we use Microsoft Yahei) as an unstyled font. All other fonts are regarded as stylized versions. During training, $\{x, y, y'\}$ represents one character in the anchor font $\mathcal{F}_0$, the same character in another font $\mathcal{F}_1$, and another character in the other font $\mathcal{F}_1$. All other training processes are the same as for learning text effects.

Then, two TET-GANs trained on font styles and text effects can constitute a uniform framework, as illustrated in Fig. 8. For an input style image, we first remove its text effects and obtain its raw text image, which is used as the reference font to adjust the input text. After that, we can use the original style image to render the text effects onto the font transfer

TABLE 3
Performance Benchmarking on the Task of General Text Effect Transfer With PSNR, SSIM, Perceptual Loss, Style Loss, and the Average Score of the User Study

Model	PSNR	SSIM	Perceptual	Style	User
AdaIN [3]	13.939	0.612	1.6147	0.0038	1.89
WCT [5]	14.802	0.619	1.8626	0.0036	1.63
Doodles [23]	18.172	0.666	1.5763	0.0031	2.98
T-Effect [9]	21.402	0.793	1.0918	0.0020	4.19
Pix2pix [10]	20.518	0.798	1.3940	0.0032	3.08
BicycleGAN [25]	20.950	0.803	1.5080	0.0033	2.63
StarGAN [12]	14.977	0.612	1.9144	0.0045	2.09
TET-GAN (ours)	**27.626**	**0.900**	**0.8250**	**0.0014**	**4.62**

The best score in each column is marked in bold, and the second best score is underlined.

result. The final output shares both the font style and text effects with the input style.

5 Experimental Results

In this section, 15 state-of-the-art style transfer models, including the proposed TET-GAN and TET-GAN+, are tested on the proposed TE141K. The models are summarized in Table 2. Through the experimental results, we provide a comprehensive benchmark for text effect transfer, analyze the performance-influencing factors of TE141K, and demonstrate the superiority of the proposed models.

5.1 Experimental Settings

Implementation Details. For state-of-the-art models, we use their public codes and default parameters. Three data-driven methods, Pix2pix [10], BicycleGAN [25], and StarGAN [12], are trained on the proposed dataset with our data preprocessing and their own data augmentation. To allow Pix2pix and BicycleGAN to handle multiple styles, we change their inputs from a single text image to a concatenation of three images: the example text effect image, its glyph counterpart and the target glyph. The architecture details of TET-GAN are provided in the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2020.2983697.

Evaluation Metrics. Since there is currently no evaluation metric specially designed for text effects, we choose to use two traditional metrics, the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM), which are widely applied for image quality assessment, and two neural-network-based metrics, perceptual loss and style loss, which are commonly used in style transfer.

The PSNR is an approximation to human perception. Let X be the image predicted by a model, and Y be the ground-truth image. The PSNR is defined as

$$\mathrm{PSNR}(X,Y) = 20 \cdot log_{10}\left(\frac{255}{||X-Y||_2^2}\right). \tag{14}$$

Compared with the PSNR, SSIM is more sensitive to changes in structural information. Given μ_x and μ_y, the average of X and Y, respectively; σ_x^2 and σ_y^2, the variance of X and Y, respectively; σ_{xy}, the covariance of X and Y; and $c_1 = 6.5025$ and $c_2 = 58.5225$, two variables to stabilize division with a weak denominator, SSIM is defined as

$$\mathrm{SSIM}(X,Y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}. \tag{15}$$

In neural style transfer [1], perceptual loss and style loss measure the semantic similarity and style similarity of two images, respectively. Let $\mathcal{F}_l$ be the feature map of the lth layer of a pretrained VGG-19 network, with Gram matrix $\mathcal{G}(\mathcal{F}) = \mathcal{F}\mathcal{F}^\top$; then, perceptual loss and style loss are

$$\begin{aligned} \mathrm{Perceptual}(X,Y) &= \sum_l ||\mathcal{F}_l(X) - \mathcal{F}_l(Y)||_2^2, \\ \mathrm{Style}(X,Y) &= \sum_l ||\mathcal{G}(\mathcal{F}_l(X)) - \mathcal{G}(\mathcal{F}_l(Y))||_2^2. \end{aligned} \tag{16}$$

(a) input (b) AdaIN (c) WCT (d) Doodles (e) T-Effect (f) pix2pix (g) BicycleGAN (h) StarGAN (i) TET-GAN

Fig. 9. Comparison with state-of-the-art methods on general text effect transfer. (a) Input example text effects with the target text in the lower-left corner. (b) AdaIN [3]. (c) WCT [5]. (d) Doodles [23]. (e) T-Effect [9]. (f) Pix2pix [10]. (g) BicycleGAN [25]. (h) StarGAN [12]. (i) TET-GAN.

TABLE 4
Performance Benchmarking on the Task of Supervised One-Reference Text Effect Transfer With PSNR, SSIM, Perceptual Loss, Style Loss, and the Average Score of the User Study

Model	PSNR	SSIM	Perceptual	Style	User
Analogy [21]	14.639	0.581	2.1202	0.0034	1.59
Doodles [23]	17.653	0.636	1.6907	0.0028	3.20
T-Effect [9]	18.654	0.712	1.4023	**0.0022**	3.96
Pix2pix+ [10]	16.656	0.660	1.7226	0.0037	2.30
TET-GAN+ (ours)	**20.192**	**0.767**	**1.4017**	0.0026	**4.26**

The best score in each column is marked in bold, and the second best score is underlined.

TABLE 5
Performance Benchmarking on the Task of Unsupervised One-Reference Text Effect Transfer With PSNR, SSIM, Perceptual Loss, Style Loss, and the Average Score of the User Study

Model	PSNR	SSIM	Perceptual	Style	User
NST [1]	11.413	0.255	2.5705	0.0045	1.56
AdaIN [3]	13.443	0.579	1.7342	0.0033	1.92
WCT [5]	13.911	0.542	2.0934	0.0033	1.93
Quilting [22]	11.045	0.354	2.6733	0.0058	1.41
CNNMRF [7]	09.309	0.337	1.8427	0.0042	1.79
UT-Effect [24]	14.877	0.609	1.7551	0.0028	2.12
TET-GAN+ (ours)	**18.724**	**0.721**	**1.4933**	**0.0027**	**3.90**

The best score in each column is marked in bold, and the second best score is underlined.

To measure the results from both local and global perspectives, we choose five layers `relu1_1`, `relu2_1`, `relu3_1`, `relu4_1` and `relu5_1`.

Furthermore, a user study was conducted in which 20 observers were asked to score the comprehensive transfer performance with 1 to 5 points (where 5 is the best). For each observer, we randomly selected 5 input pairs and asked them to score the corresponding results of all models. Finally, we collected 2,000 scores and calculated the average score for each model on each task.

5.2 Benchmarking on General Text Effect Transfer

We first benchmark models on general text effect transfer, where text styles in the training and testing phases are the same. As shown in Table 3, TET-GAN performs the best on all metrics. Representative results are shown in Fig. 9.

Two representative global models, AdaIN [3] and WCT [5], fail to perform well in both quantity and quality. AdaIN fails to reconstruct details and has color deviations, while WCT creates wavy artifacts and mingles the foreground and background. This may be because AdaIN and WCT are designed to model the style as global statistics, and this representation is not suitable for text effect transfer.

We select two representative local models, Doodles [23] and T-Effect [9]. Doodles fails to preserve global structure and suffers from artifacts. In Table 3, T-Effect performs second best on four metrics. However, since the T-Effect processes patches in the pixel domain, it causes obvious color and structure discontinuity. In addition, when the edge of the input glyph differs greatly from the target glyph, T-Effect fails to adapt well to the new glyph.

(a) input (b) Analogy (c) Doodles (d) T-Effect (e) Pix2pix+ (f) TET-GAN+

Fig. 10. Comparison with other methods on one-reference supervised text effect transfer. (a) Input example text effects with the target text in the lower-left corner. (b) Analogy [21]. (c) Doodles [23]. (d) T-Effect [9]. (e) Pix2pix+ [10]. (f) TET-GAN+.

Regarding GAN-based methods, both Pix2pix [10] and BicycleGAN [25] achieve a PSNR of over 20 and an SSIM of over 0.79. However, they cannot eliminate the original input, leaving some ghosting artifacts. StarGAN [12] learns some color mappings but fails to synthesize texture details and suffers from distinct checkerboard artifacts, which also leads to low quantitative performance. In summary, GAN-based methods can largely realize text effect transfer; however, the visual effect is not satisfactory, which may be due to the instability of GAN and limited network capability.

By comparison, our network learns valid glyph features and style features, thus precisely transferring text effects while preserving the glyph well.

5.3 Benchmarking on Supervised One-Reference Text Effect Transfer

In the task of supervised one-reference text effect transfer, only one observed example pair is provided. In addition to existing methods, we select Pix2pix [10], which has the best user score among the GAN-based models in the last task, and apply our one-reference learning strategy proposed in Section 4.2 to it, making Pix2pix able to process unseen text effects. This variant is named Pix2pix+.

As shown in Table 4, the proposed TET-GAN+ achieves the best results on PSNR, SSIM, content loss and the user study, and performs second best on style loss. T-Effect [9] performs slightly better than TET-GAN+ on style loss. This may be because style loss mainly focuses on local similarity, and the T-effect can easily achieve good local similarity by directly copying and fusing patches from the source image. It can be observed in Fig. 10 that the T-Effect is less able to preserve shape, texture regularity and color continuity. In terms of efficiency, the released T-Effect implemented in MATLAB requires approximately 150 s per image with an Intel Xeon E5-1620 CPU (no GPU version available). In comparison, our feed-forward method only takes approximately 0.33 s per image with an Intel Core i7-6850K CPU and 10 ms per image with a GeForce GTX 1080 GPU after a three-minute finetuning.

The other three methods are not as good as TET-GAN+ in terms of either quantitative or qualitative performance. Two local methods face problems: Analogy [21] cannot well preserve the shape of the glyph, while Doodles [23] causes artifacts and distorts textures. Although, taking advantage of the proposed finetuning strategy, Pix2pix+ [10] can transfer

Fig. 11. Comparison with other methods on one-reference unsupervised text effect transfer. (a) Input example text effects with the target text in the lower-left corner. (b) NST [1]. (c) AdaIN [3]. (d) WCT [5]. (e) Quilting [22]. (f) CNNMRF [7]. (g) UT-Effect [24]. (h) TET-GAN+.

basal colors to the target glyph, it suffers from severe detail loss. Compared with Pix2pix+, TET-GAN+ can adapt to and reconstruct new textures due to feature disentanglement.

5.4 Benchmarking on Unsupervised One-Reference Text Effect Transfer

The task of unsupervised one-reference text effect transfer is challenging; only one observed example is provided for the models. The advantages of our approach are more pronounced. As shown in Table 5, TET-GAN+ achieves the best result on all five metrics. Representative results are illustrated in Fig. 11.

Three global methods, NST [1], AdaIN [3] and WCT [5], all fail to produce satisfactory quantity and quality results. NST cannot correctly find the correspondence between the texture and glyph, leading to severe shape distortion and interwoven textures. AdaIN and WCT encounter the same problem as in Section 5.2. Concerning local methods, CNNMRF [7] fails to adapt text effect features to the target glyph. Quilting [22] and UT-Effect [24] distort glyphs. Taking full advantage of destylization, TET-GAN+ can distinguish the foreground and background well, and therefore can successfully reconstruct the global structure.

Compared with Table 4, we find that the quantitative performance in Table 5 is much worse, which indicates that the unsupervised task is more difficult than the supervised one. This is obvious since the models are provided with much less information. However, the proposed TET-GAN+ still achieves satisfactory results, which demonstrates that the unsupervised task is solvable, and the proposed unsupervised one-reference training scheme is feasible.

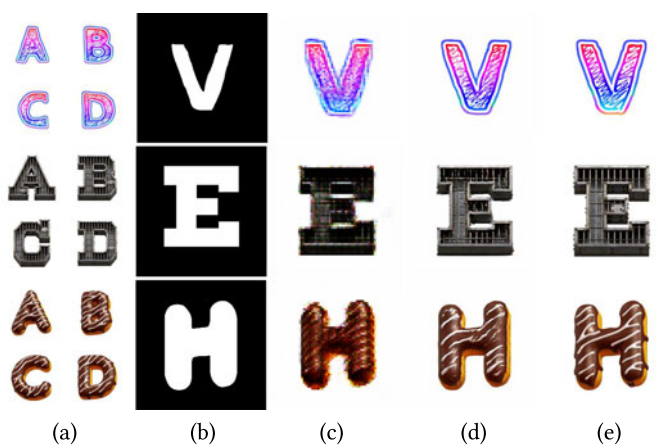


Fig. 12. Comparison with MC-GAN [17] on TE141K-F. (a) Input style images for training. (b) Target glyph images. (c) Results of MC-GAN. (d) Results of TET-GAN+. (e) Ground-truth.

5.5 Comparision With MC-GAN

MC-GAN [17] is a few-reference method and can only handle 26 capital letters with a small image resolution of 64 × 64. Moreover, MC-GAN only supports text effects with white backgrounds. Therefore, to compare MC-GAN, we need to build a new testing set.

We first collect 10 styles from FlamingText,[2] the same website used by MC-GAN. Note that FlamingText renders text effects in an automatic way, which is similar to our data collection method using PhotoShop batch tools. To fairly examine the ability of a model to handle unexpected text effects, we collect 10 more challenging data from Handmadefont,[3] an online shop of professionally designed handmade or 3D text effects. Finally, we collect 20 styles with 5 images for each style, summing up to 100 text effect images, and manually label the corresponding glyph for each style image. This testing set is named *TE141K-F* ('F' for few-reference).

Fig. 12 shows the comparison results where four characters are provided for few-reference learning and one for testing. We directly feed MC-GAN with the ground-truth glyph images to make a fair comparison with TET-GAN. As can be seen, the text effects in TE141K-F have quite challenging structures and textures, which are not reconstructed by MC-GAN. In Fig. 12d, TET-GAN+ generates high-quality results with rich textures, and synthesizes complex but plausible structures. However, compared to the ground-truth, the

2. http://www6.flamingtext.com/All-Logos
3. https://handmadefont.com/

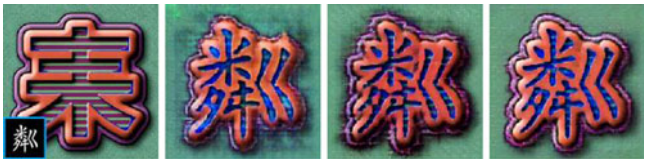


Fig. 13. Effect of autoencoder loss and feature loss. (a) Input. (b) Model without $\mathcal{L}_{\text{gly}}$ and $\mathcal{L}_{\text{dfeat}}$. (c) Model without $\mathcal{L}_{\text{dfeat}}$. (d) Full model.

details are still slightly distorted and mixed in the results of TET-GAN+. This indicates substantial room for improvement in terms of few-reference learning. Full results on TE141K-F are provided in the supplementary material, available online.

5.6 Analysis of TE141K

We further analyze the performance-influencing factors of TE141K with the benchmarking results. To quantify the difficulty of transferring a certain text style, we use its average user score value over eight transfer models benchmarked on the task of general text effect transfer to represent its transferring difficulty.

To explore the factors that determine the difficulty of transferring text effects, we first use one-hot encoding to represent whether text effects contain the 24 subclasses introduced in Section 2.3. Then, we adopt linear regression to fit the average user score. The Pearson correlation coefficient of this regression is 0.742. The linear regression weights can indicate whether a subclass plays an important role in determining the user scores. The three largest positive weight values are *One-Side Stroke* (0.47798), *Background Hard Texture* (0.39513), and *Foreground Normal Texture* (0.37166). The reason for this is twofold. First, these distinct visual elements are easily perceived by human eyes to influence user decisions. Second, these visual elements are mostly anisotropic and irregular, which are difficult for transfer models to characterize and generate. Additionally, *Normal Stroke* (-1.34573) has the lowest negative weight values, while the values of *Thick Stroke* (-0.73353) and *Thin Stroke* (-0.60621) are also negative. This is because with the help of glyph shape and data preprocessing to provide distance information, strokes become easy to model and reconstruct. In conclusion, generating irregular textures or shapes around glyphs constitutes the major challenge of text effect transfer. This suggests focusing on modeling irregular textures or shapes in subsequent studies.

5.7 Performance Analysis of TET-GAN

In addition to model benchmarking, we conducted experiments to further analyze the performance of TET-GAN.

Ablation Study. In Fig. 13, we study the effect of the reconstruction loss (Eq. (4)) and the feature loss (Eq. (2)). Without these two losses, even the color palette of the example style is not correctly transferred. In Fig. 13c, the glyph is not fully disentangled from the style, leading to bleeding artifacts. The satisfying results in Fig. 13d verify that our feature loss effectively guides TET-GAN to extract valid content representations to synthesize clean text effects. For one-reference text effect transfer, as shown in Fig. 14, if trained from scratch (namely, the self-supervised adversarial training scheme [31]), the performance of our network drops dramatically, verifying that pretraining on our dataset successfully teaches our network the domain knowledge of text effect synthesis.

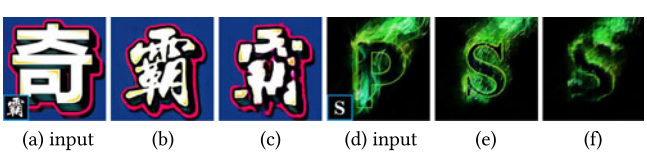
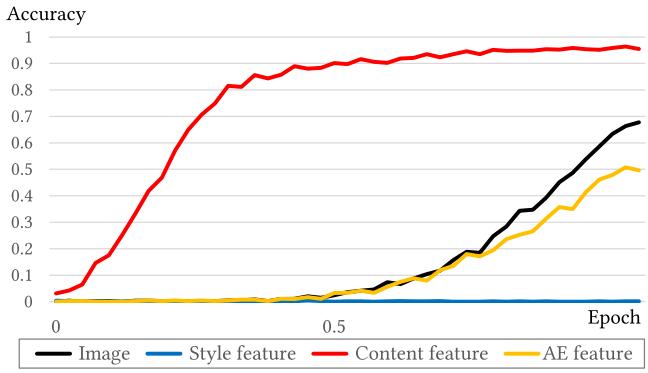
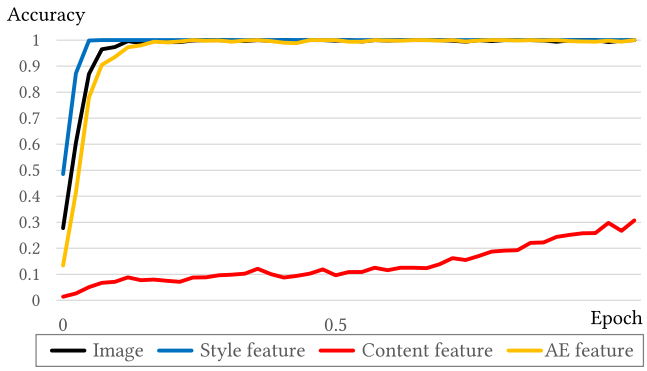


Fig. 14. Effect of pretraining for one-reference text effect transfer. (a), (d): Input. (b), (e): Our results. (c), (f): Results without pretraining.



(a) Classification accuracy of content during training



(b) Classification accuracy of styles during training

Fig. 15. Analysis of the disentangled style feature and content feature in text effect classification and glyph classification.

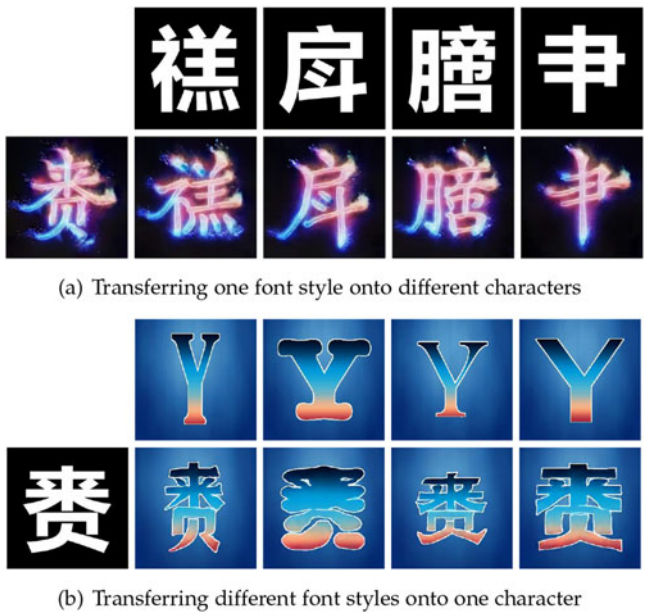
Disentanglement of Style and Content Features. TET-GAN disentangles content and style to enable multi-style transfer and removal. To measure the entanglement of the two entities, we design experiments to analyze the representation capabilities of the extracted features. Based on the intuition that a meaningful style (content) feature contains enough information to represent its style (content) but little information to represent its content (style), we perform classification over both features. Specifically, we extract the content feature and style feature of a style image as the output features of the sharing layers of $G_{\mathcal{X}}$ and $G_{\mathcal{Y}}$, respectively. We additionally train an autoencoder over style images to extract the feature that preserves both the glyph and style information, which we denote as the AE feature for comparison. Then, we train a five-layer classification network to predict the content and style label based on the input content feature, style feature and AE feature. Another six-layer classification network fed with the original style image is also

Fig. 16. Stylization results in different font styles. First row: input text images. Next rows: input style images and the style transfer results.

trained as a reference. We elaborate the network architecture to make the two classification networks have similar numbers of parameters. The prediction accuracies during training over TE141K-C are plotted in Fig. 15.

The AE feature is comparable to the image input. Compared with other inputs, the content feature extracted by TET-GAN contains highly representative and discriminative glyph information but very little style information. The same is true for the style feature, verifying the disentanglement of content and style. This is because, for the content feature, E^c_y is tasked with approaching the ideal glyph feature via our feature loss (Eq. (4)). For the style feature, E^s_y aims to extract style features that apply to arbitrary glyphs in content images, and is thus driven to eliminate the irrelevant glyph information from the style image. In addition to the drive of the loss functions, another reason may be the good nature of our collected dataset, that is, high quality paired data with style labels.

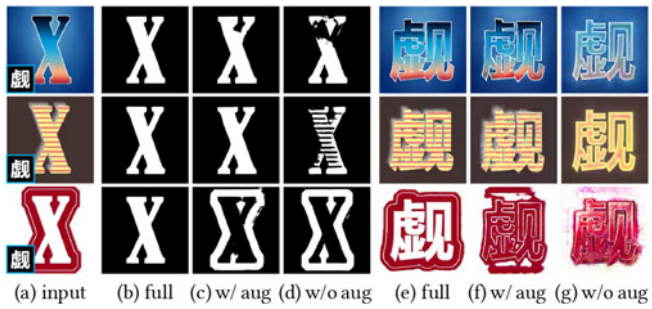
Generalizability Across Font. To study the generalizability of TET-GAN across font styles, we select several text images in quite different fonts, as shown in Fig. 16. Many of them are handwriting styles with irregular strokes. TET-GAN



(a) Transferring one font style onto different characters

(b) Transferring different font styles onto one character

Fig. 17. Joint font style and text effect transfer results. (a) First row: input text images. Second row: input style image and the style transfer results. (b) First row: input style images. Second row: input text image and the style transfer results.



(a) input (b) full (c) w/ aug (d) w/o aug (e) full (f) w/ aug (g) w/o aug

Fig. 18. Destylization and stylization results of TET-GAN w/ and w/o adversarial augmentation. (a): Input example text effects from TE141K-E with the target text in the lower-left corner. (b)-(g): Destylization and stylization results by the models trained with supervision on the full TE141K (full), trained with supervision on TE141K-C and adversarially augmented with TE141K-E (w/ aug) and trained with only supervision on TE141K-C (w/o aug), respectively.

successfully renders plausible text effects on these challenging strokes, verifying its capability of transferring the text effects on a reference image to other glyphs in different font styles.

Joint Font Style and Text Effect Transfer. In Fig. 17, we study the performance of TET-GAN in joint font style and text effect transfer. We prepare a font dataset with 775 Chinese characters in 30 different font styles and train TET-GAN with the first 740 characters. The style transfer results over the remaining unseen characters are shown in Fig. 17a. It can be seen that the font style and text effects are effectively transferred. In Fig. 17b, we use English letters in TE141K-E as a reference style. Note that both the glyph and the font styles are unseen during the training of our font style transfer model. TET-GAN selects the best-matched font among the 30 training fonts based on the reference image to render the target text, which achieves satisfactory style consistency with the input.

Semisupervised Text Effect Transfer. In Fig. 18, we compare the performance of supervised text transfer with and without adversarial augmentation. We manually divide TE141K-E into two parts with no overlap in glyphs to generate our unpaired data. The model trained only on TE141K-C serves as a baseline to show the performance of our model on unseen glyphs and styles. Meanwhile, the model trained on the full TE141K gives an upper bound that we can expect from unsupervised learning. It can be seen that our semisupervised model learns from unpaired data to better imitate the target styles. We also observe a clear improvement in the destylization of English glyphs that is fairly different from that of the training data in TE141K-C, indicating better glyph generalization of the model. However, our semisupervised

Fig. 19. Applications of TET-GAN for style interpolation.

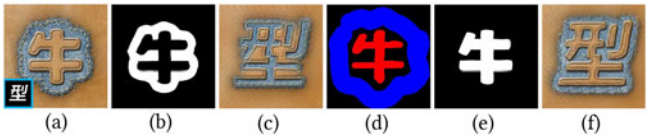


Fig. 20. User-interactive unsupervised style transfer. (a): Example text effects and the target text. (b), (c): Our destylization and stylization results after finetuning. (d): A mask provided by the user, where the blue and red regions indicate the background and foreground, respectively. (e), (f): Our destylization and stylization results with the help of the mask.

model fails to handle the challenging text effects in the third row of Fig. 18, verifying that our dataset is challenging and there is still much room for improvement in semisupervised learning.

Style Interpolation. The flexibility of TET-GAN is further shown by the application of style interpolation. The explicit style representations enable intelligent style editing. Fig. 19 shows an example of style fusion. We interpolate between four different style features and decode the integrated features back to the image space, obtaining new text effects.

Failure Case and User Interaction. While our approach has generated appealing results, some limitations still exist. Our destylization subnetwork is not fool-proof due to the extreme diversity of the text effects, which may differ completely from our collected text effects. Fig. 20 shows a failure case of one-reference unsupervised text effect transfer. Our network fails to recognize the glyph. As a result, in the stylization result, the text effects in the foreground and background are reversed. This problem can possibly be solved by user interaction. Users can simply paint a few strokes (Fig. 20d) to provide a priori information about the foreground and the background, which is then fed into the network as guidance to constrain glyph extraction. Specifically, let M_f and M_g be the binary mask of foreground and background (i.e., the red channel and the blue channel in Fig. 20d) provided by the user, respectively. Then, a guidance loss is added to Eq. (1)

$$\begin{aligned}\mathcal{L}_{\text{guid}} = \mathbb{E}_y[\|G_{\mathcal{X}}(E^c_{\mathcal{Y}}(y)) \odot M_f - M_f\|_1] \\ + \mathbb{E}_y[\|G_{\mathcal{X}}(E^c_{\mathcal{Y}}(y)) \odot M_b - \mathbf{0}\|_1],\end{aligned} \tag{17}$$

where $\odot$ is the elementwise multiplication operator. As shown in Fig. 20e, under the guidance of $\mathcal{L}_{\text{guid}}$, the glyph is correctly extracted, and the quality of the style transfer result (Fig. 20f) is thereby greatly improved.

6 Conclusion

In this paper, we introduce a novel text effects dataset with 141K text effect/glyph pairs in total, which consists of 152 professionally designed text effects and 3K different kinds of glyphs, including English letters, Chinese characters, Japanese kana, and Arabic numerals. Statistics and experimental results validate the challenges of the proposed dataset. In addition, we design effective data preprocessing and augmentation methods that can improve the robustness of transfer models. Moreover, we present a novel TET-GAN for text effect transfer. We integrate stylization and destylization into one uniform framework to jointly learn valid content and style representations of the artistic text. The benchmarking results demonstrate the superiority of TET-GAN in generating high-quality artistic typography. As a future direction, one may explore other more sophisticated style editing methods on the proposed dataset, such as background replacement, color adjustment and texture attribute editing. We believe the proposed dataset has the potential to boost the development of corresponding research areas.

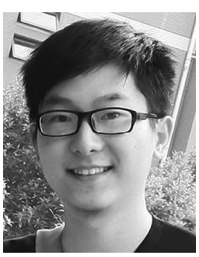
Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under contract No.61772043, in part by Beijing Natural Science Foundation under contract No.L182002. Shuai Yang and Wenjing Wang contributed equally to this work.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2414–2423.
[2] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–11.
[3] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 1510–1519.
[4] Y. Li, N. Wang, J. Liu, and X. Hou, "Demystifying neural style transfer," in *Int. Joint Conf. Artif. Intell.*, 2017, pp. 2230–2236.
[5] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 386–396.
[6] L. A. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 262–270.
[7] C. Li and M. Wand, "Combining Markov random fields and convolutional neural networks for image synthesis," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2479–2486.
[8] T. Q. Chen and M. Schmidt, "Fast patch-based style transfer of arbitrary style," 2016, [Online]. Available: https://arxiv.org/abs/1612.04337
[9] S. Yang, J. Liu, Z. Lian, and Z. Guo, "Awesome typography: Statistics-based text effects transfer," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7464–7473.
[10] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5967–5976.
[11] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2242–2251.
[12] Y. Choi, M. Choi, M. Kim, J. W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8789–8797.
[13] R. Tyleček and R. Šára, "Spatial pattern templates for recognition of objects with regular structure," in *Proc. German Conf. Pattern Recognit.*, 2013, pp. 364–374.
[14] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros, "What makes paris look like paris?" *ACM Trans. Graph.*, vol. 31, no. 4, pp. 101:1–101:9, 2012.
[15] A. Yu and K. Grauman, "Fine-grained visual comparisons with local learning," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 192–199.
[16] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 597–613.
[17] S. Azadi, M. Fisher, V. Kim, Z. Wang, E. Shechtman, and T. Darrell, "Multi-content GAN for few-shot font style transfer," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7564–7573.
[18] S. Yang, J. Liu, W. Wang, and Z. Guo, "TET-GAN: Text effects transfer via stylization and destylization," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 1238–1245.
[19] L. V. D. Maaten and G. Hinton, "Visualizing data using T-SNE," *J. Mach. Learn. Res.*, vol. 9, no. Nov., pp. 2579–2605, 2008.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–10.
[21] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proc. Conf. Comput. Graph. Interactive Techn.*, 2001, pp. 327–340.
[22] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proc. ACM Conf. Comput. Graph. Interactive Techn.*, 2001, pp. 341–346.
[23] A. J. Champandard, "Semantic style transfer and turning two-bit doodles into fine artworks," 2016. [Online]. Available: https://arxiv.org/abs/1603.01768
[24] S. Yang, J. Liu, W. Yang, and Z. Guo, "Context-aware text-based binary image stylization and synthesis," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 952–964, Feb. 2019.
[25] J.-Y. Zhu *et al.*"Toward multimodal image-to-image translation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 465–476.
[26] J. Johnson, A. Alahi, and F. F. Li, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.
[27] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky, "Texture networks: Feed-forward synthesis of textures and stylized images," in *Proc. IEEE Int. Conf. Mach. Learn.*, 2016, pp. 1349–1357.
[28] X. Wang, G. Oxholm, D. Zhang, and Y. F. Wang, "Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7178–7186.
[29] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "StyleBank: An explicit representation for neural image style transfer," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2770–2779.
[30] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, 2015, pp. 234–241.
[31] Y. Zhou, Z. Zhu, X. Bai, D. Lischinski, D. Cohen-Or, and H. Huang, "Non-stationary texture synthesis by adversarial expansion," *ACM Trans. Graph.*, vol. 37, no. 4, 2018, Art. no. 49.
[32] E. Simo-Serra, S. Iizuka, and H. Ishikawa, "Mastering sketching: Adversarial augmentation for structured prediction," *ACM Trans. Graph.*, vol. 37, no. 1, 2018, Art. no. 11.



Shuai Yang (Student Member, IEEE) received the BS degree in computer science from Peking University, Beijing, China, in 2015. He is currently working toward the PhD degree in the Wangxuan Institute of Computer Technology, Peking University, Beijing, China. His current research interests include image stylization and image inpainting.

Wenjing Wang (Student Member, IEEE) received the BS degree in data science from Peking University, Beijing, China, in 2019. She is currently working toward the master's degree in the Wangxuan Institute of Computer Technology, Peking University, Beijing, China. Her current research interests include image stylization, synthesis, and enhancement.

Jiaying Liu (Senior Member, IEEE) received the PhD (Hons.) degree in computer science from Peking University, Beijing, China, in 2010. She is currently an associate professor with the Wangxuan Institute of Computer Technology, Peking University. She has authored more than 100 technical articles in refereed journals and proceedings, and holds 42 granted patents. Her current research interests include multimedia signal processing, compression, and computer vision. She is also a senior member of CSIG and CCF. She was a visiting scholar with the University of Southern California, Los Angeles, from 2007 to 2008. She was a visiting researcher with the Microsoft Research Asia in 2015 supported by the Star Track Young Faculties Award. She has served as a member of Membership Services Committee in IEEE Signal Processing Society, a member of Multimedia Systems & Applications Technical Committee (MSA TC), Visual Signal Processing and Communications Technical Committee (VSPC TC) in IEEE Circuits and Systems Society, a member of the Image, Video, and Multimedia (IVM) Technical Committee in APSIPA. She has also served as the associate editor of the *IEEE Transactions on Image Processing*, and the *Elsevier Journal of Visual Communication and Image Representation*, the technical program chair of IEEE VCIP-2019/ACM ICMR-2021, the publicity chair of IEEE ICME-2020/ICIP-2019, and the area chair of CVPR-2021/ECCV-2020/ICCV-2019. She was the APSIPA distinguished lecturer (2016-2017).