TextCtrl: Diffusion-based Scene Text Editing with Prior Guidance Control

Weichao Zeng[1,3] **Yan Shu**[1] **Zhenhang Li**[1,3]
Dongbao Yang[1,3] **Yu Zhou**[2*]
[1] Institute of Information Engineering, Chinese Academy of Sciences
[2] VCIP & TMCC & DISSec, College of Computer Science, Nankai University
[3] School of Cyber Security, University of Chinese Academy of Sciences
{zengweichao, lizhenhang, yangdongbao}@iie.ac.cn
shuyan9812@gamil.com, yzhou@nankai.edu.cn

Abstract

Centred on content modification and style preservation, Scene Text Editing (STE) remains a challenging task despite considerable progress in text-to-image synthesis and text-driven image manipulation recently. GAN-based STE methods generally encounter a common issue of model generalization, while Diffusion-based STE methods suffer from undesired style deviations. To address these problems, we propose **TextCtrl**, a diffusion-based method that edits text with prior guidance control. Our method consists of two key components: (i) By constructing fine-grained text style disentanglement and robust text glyph structure representation, TextCtrl explicitly incorporates Style-Structure guidance into model design and network training, significantly improving text style consistency and rendering accuracy. (ii) To further leverage the style prior, a Glyph-adaptive Mutual Self-attention mechanism is proposed which deconstructs the implicit fine-grained features of the source image to enhance style consistency and vision quality during inference. Furthermore, to fill the vacancy of the real-world STE evaluation benchmark, we create the first real-world image-pair dataset termed **ScenePair** for fair comparisons. Experiments demonstrate the effectiveness of TextCtrl compared with previous methods concerning both style fidelity and text accuracy. Project page: https://github.com/weichaozeng/TextCtrl.

1 Introduction

Scene Text Editing (STE) refers to modifying the text with desired content on an input image while preserving the styles and textures of both the text and the background to maintain a realistic appearance [1]. As a newly emerging task in the field of scene text processing [2], STE not only possesses distinctive application value [3, 4, 5] but also benefits the text-oriented downstream research in detection [6], recognition [7, 8], spotting [9] and reasoning [10, 11]. Recently, increasing attention has been paid to GAN-based and diffusion-based scene text editing methods.

Exploiting the Generative Adversarial Networks (GANs) [12], early works [1, 13] decompose STE into three subtasks: foreground text style transfer, background restoration and fusion. The divide-and-conquer manner significantly reduces the difficulty of pattern learning and enables the pre-training of sub-modules with additional supervision [14]. However, the generalization capabilities of these methods are inevitably limited due to the constrained model capacity of GANs [15] and the challenges in accurately decomposing text styles [4]. Besides, as observed in experiments, the divide-

[*] Corresponding Author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-4396

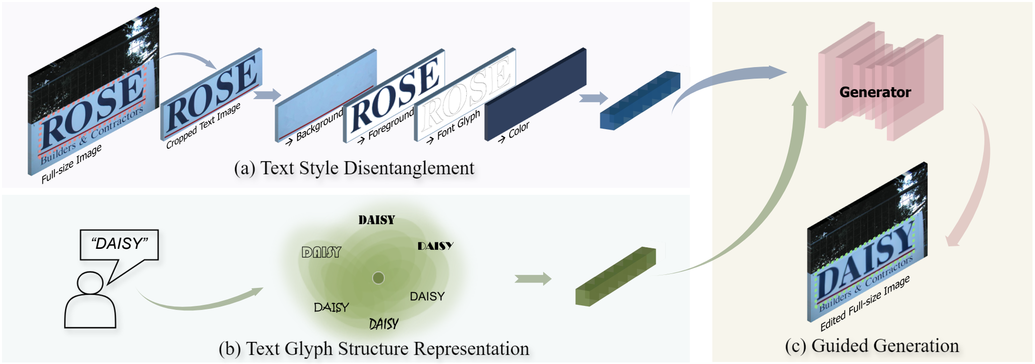


Figure 1: Conceptual illustration of the decomposition of STE by TextCtrl. (a) Text style is disentangled into text background, text foreground, text font glyph and text color features. (b) Text glyph structure is represented by the cluster centroid of various font text features. (c) The explicit style features and structure features guide the generator to perform scene text editing.

and-conquer design brings about the "bucket effect", wherein the unstable background restoration quality leads to messy fusion artifacts.

Recently, large-scale text-to-image diffusion models [16, 17] have convincingly demonstrated strong capabilities in image synthesis and processing. Several methods attempt to realize STE in a conditional synthesis manner, including style image concatenation [18] and one-shot style adaptation [19]. However, these methods are limited to **the coarse-grained learning of miscellaneous styles** from text images. Other methods [20, 21, 22] tend to resolve STE in a universal framework along with STG (Scene Text Generation) in an inpainting manner conditioned on full images, which enables the leverage of large-scale data for self-supervised learning [23]. Nevertheless, their **style guidance predominantly originates from the image's unmasked regions**, which can be unreliable in complex scenarios and fail in style consistency. Besides, resulting from **the weak correlation between text prompt and glyph structure** [24, 25, 26], diffusion-based STE methods are prone to generating typos, which decreases the text rendering accuracy.

For the aforementioned problems, we identify insufficient prior guidance on both style and structure as the primary factor that impedes the previous methods from performing accurate and faithful scene text editing. As depicted in Fig. 1, we propose a conditional diffusion-based STE model, wherein our method decomposes the prerequisite of STE into two main aspects: text style disentanglement and text glyph representation. The fine-grained disentangled text style features ensure visual coherency, while the robust glyph structure representation improves text rendering accuracy. The dual Style-Structure guidance collectively contributes to significant enhancements in STE performance.

Furthermore, undesired color deviation and texture degradation compared with the source text image occasionally occur in the inference of diffusion-based STE methods, which is attributed to the error accumulation in the denoising process [27] as well as the domain gap between training and inference [19]. To overcome this limitation, we introduce a glyph-adaptive mutual self-attention mechanism to improve the generator, which sets up a parallel reconstruction branch to introduce the source image style prior through cross-branch integration. The refined sampling process effectively eliminates visual inconsistency without requiring additional tuning.

Additionally, the deficiency of real-world evaluation benchmarks on STE has become a non-negligible problem as increasing methods are proposed. Early assessments [1], which rely on synthetic data, face significant limitations in practice due to the domain gap. Recent evaluations [14] emphasize text accuracy in edited real images but fail to benchmark visual quality adequately. Based on the observation that scene texts often occur in phrases with the same style and background in real-world scenery, we elaborately collect 1,280 text image pairs in terms of similar style and word length from scene text datasets to build the **ScenePair** dataset enabling comprehensive evaluation.

In summary, we improve STE with the full leverage of **Text** prior for comprehensive guidance **Control** throughout the model design, network training and inference control, termed as **TextCtrl**. Our main contributions are as follows:

- For the first time, we decompose the prerequisite of STE into fine-grained style disentanglement as well as glyph structure representation and incorporate the Style-Structure guidance with diffusion models to improve rendering accuracy and style fidelity.
- For further style coherency control during sampling, with the leverage of additional prior guidance through the reconstruction of the source image, we introduce a glyph-adaptive mutual self-attention mechanism that effectively eliminates visual inconsistency.
- We propose an evaluation benchmark **ScenePair** consisting of cropped text image pairs along with original full-size images. To the best of our knowledge, it is the first pairwise real-world dataset for STE which enables both visual quality assessment and rendering accuracy evaluation.

2 Related work

GAN-based Scene Text Editing. SRNet [1] first introduces the word-level editing method built in a divide-and-conquer manner. SwapText [13] further enhances SRNet with Thin Plate Spline Interpolation Network for curved text modification while STRIVE [28] extends the framework into the video domain of scene text replacement. Besides, TextStyleBrush [4] adopts a self-supervised strategy building on StyleGAN2 [29] while MOSTEL [14] designs a semi-supervised training scheme.

Diffusion-based Scene Text Editing. Numerous studies have focused on adapting the diffusion model for scene text manipulation. DiffSTE [20] improves pre-trained diffusion models with a dual encoder design, wherein a character encoder for render accuracy and an instruction encoder for style control is used. DiffUTE [23] further utilizes an OCR-based image encoder as an alternative to CLIP Text encoder. Moreover, TextDiffuser [21] and UDiffText [30] leverage character segmentation masks for condition input and supervised labels respectively. To leverage text style, LEG [18] concats the source image as input while DBEST [19] relies on a fine-tuning process during inference. Recently, AnyText [22] adopted a universal framework to resolve STE and STG in multiple languages based on the prevalent ControlNet [31].

Image Editing with Diffusion Models. Image editing aims to manipulate a certain attribute (e.g. color, posture, position) of the target object while keeping the other context unchanged, which can be seen as the parent task of STE. Recent Diffusion-based methods have shown unprecedented potential with a wide variety of designs. Model-tuning methods [32, 33] fine-tune the entire model to enhance subject embedding in the output domain. Leveraging DDIM inversion [34], prompt-tuning methods [27] turn to improve identity preservation by optimizing null-text prompts through classifier-free guidance sampling [35]. Recently, [36, 37] explored the self-attention layers in LDMs and demonstrated the rich semantic information preserved in queries, keys and values. Through the cross-frame substitute of keys and values of self-attention, they perform non-rigid editing without additional tuning. [38] further extends the cross-frame interaction to video domain for motion editing.

3 Method

Based on a conditional synthesis manner, in this work, we define the scene text editing process as $I_{edit} = \mathcal{G}(C_{struct}, C_{style})$ as shown in Fig. 2 (c). The text glyph structure feature is acquired from a character-based structure encoder as $C_{struct} = \mathcal{T}(C_{text})$ in Fig. 2 (a) and the text style feature is derived from a style encoder $C_{style} = \mathcal{S}(I_{source})$ in Fig. 2 (b). The module design and pre-training strategy to enable precise extraction for glyph structure and fine-grained disentanglement of text style are introduced in section 3.1 and section 3.2 respectively, with the whole model training process illustrated in section 3.3. Furthermore, details of the improved tuning-free inference control and the proposed glyph-adaptive mutual self-attention mechanism in Fig. 2 (d) are presented in section 3.4.

3.1 Text Glyph Structure Representation Pre-training

Distinctive from natural objects, scene text possesses a complicated non-convex structure, wherein a minor stroke discrepancy can significantly alter visual perception and lead to misinterpretation [39], thus presenting unique challenges to editing accuracy. For scene text editing, an ideal text encoder is capable of encoding the target text concerning glyph structure rather than semantic information

https://doi.org/10.52202/079017-4396

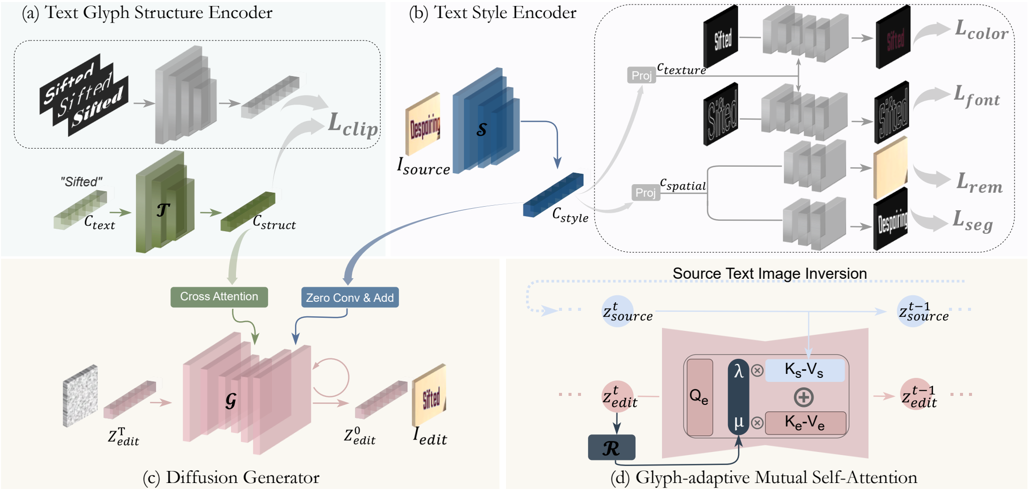


Figure 2: Decomposed framework of TextCtrl. (a) Text glyph structure encoder $\mathcal{T}$ with corresponding glyph structure representation pre-training. (b) Text style encoder $\mathcal{S}$ with corresponding style disentanglement pre-training. (c) Prior guided diffusion generator $\mathcal{G}$. (d) The improved inference control with the Glyph-adaptive Mutual Self-attention mechanism.

[20, 21] or certain image template [14, 23]. Specifically, for a certain text $C_{text} =$ "$Sifted$", encoder $\mathcal{T}$ is expected to be aware of the glyph structure of "S", "i", "f", "t", "e", "d" respectively.

To this end, we adopt a character-level text encoder to align the target text feature with its visual glyph structure. As depicted in Fig. 2 (a), the target text embedding in character level is processed with a transformer encoder $\mathcal{T}$ to generate glyph structure features $C_{struct} \in \mathbb{R}^{L \times d}$, which is further aligned to the visual feature of corresponding text image extracted by a frozen pre-trained scene text recognizer with CLIP loss [30, 40] $\mathcal{L}_{clip}$. Differ from [30], we collect vast quantities of text fonts constructing a cluster $\{font_1, font_2...font_n\}$ to render the corresponding text image with diverse fonts during training. The font-variance augmentation brings continuous glyph structure variation which implicitly enhances the projection from C_{struct} to the cluster centroids of visual features for robust text glyph structure representation.

3.2 Text Style Disentanglement Pre-training

Text styles comprise a variety of aspects, including font, color, spatial transformation and stereoscopic effect, which visually mingle with each other and bring obstacles to disentangle the style features precisely in previous works. To realize the fine-grained disentanglement of the text style, we propose a multi-task pre-training paradigm as illustrated in Fig. 2 (b), involving text color transfer, text font transfer, text removal and text segmentation.

Concretely, given a text image $I_{source} \in \mathbb{R}^{3 \times H \times W}$, we first extract the style feature $C_{style} \in \mathbb{R}^{N \times d}$ with a ViT [41] backbone $\mathcal{S}$, which is projected to texture feature $c_{texture} \in \mathbb{R}^{N \times d}$ and spatial feature $c_{spatial} \in \mathbb{R}^{N \times d}$ respectively. Subsequently, $c_{texture}$ is employed in text color transfer and text font transfer while $c_{spatial}$ is utilized for text removal and text segmentation.

Text Color Transfer. Since both intrinsic style and lighting conditions determine the text color, it is challenging to label or classify the holistic color. Instead, we refer to image style transfer and implicitly extract color through colorization training. A light-weight encoder-decoder $\mathcal{F}^c$ is built to provide colorization on a black and white text image $i_{in}^c \in \mathbb{R}^{1 \times h \times w}$ with an Adaptive Instance Normalization [42] $\mathcal{A}$ for source text color image $i_{out}^c \in \mathbb{R}^{3 \times h \times w}$ written as:

$$i_{out}^c = \mathcal{F}_{dec}^c(\mathcal{A}(\mathcal{F}_{enc}^c(i_{in}^c), c_{texture})), \tag{1}$$

Text Font Transfer. With a common intention with color transfer to capture stylized information but focusing on glyph boundary, font transfer is realized through the boundary reshaping process. Another light-weight encoder-decoder $\mathcal{F}^f$ is employed to transfer a template font text glyph $i^f_{in} \in \mathbb{R}^{3\times h\times w}$ to the source font text glyph $i^f_{out} \in \mathbb{R}^{3\times h\times w}$ through Pyramid Pooling Module [43] $\mathcal{P}$ in latent space as:

$$i^f_{out} = \mathcal{F}^f_{dec}(\mathcal{P}(\mathcal{F}^f_{enc}(i^f_{in}), c_{texture})), \tag{2}$$

Text Removal and Text Segmentation. Text removal aims at erasing the text pixels and reasoning the background pixels covered by text while text segmentation decouples the spatial relationships between background and text. A residual convolution block with spatial attention mechanism [44] is adopted to construct a removal head $\mathcal{F}^r$ and a segmentation head $\mathcal{F}^s$ respectively to generate predicted background $i^r_{out} \in \mathbb{R}^{3\times h\times w}$ and predicted mask $i^s_{out} \in \mathbb{R}^{1\times h\times w}$ as:

$$i^r_{out} = \mathcal{F}^r(c_{spatial}), \quad i^s_{out} = \mathcal{F}^s(c_{spatial}), \tag{3}$$

Multi-task Loss. With the multi-task pre-training for fostering the text style extraction and disentanglement ability of SE, the whole loss function for style pre-training can be expressed as:

$$\mathcal{L}_{disentangle} = \mathcal{L}_{color}(i^c_{out}, i^c_{gt}) + \mathcal{L}_{font}(i^f_{out}, i^f_{gt}) + \mathcal{L}_{rem}(i^r_{out}, i^r_{gt}) + \mathcal{L}_{seg}(i^s_{out}, i^s_{gt}), \tag{4}$$

wherein we leverage MSE loss for $\mathcal{L}_{color}$, MAE loss for $\mathcal{L}_{rem}$ and Dice loss [45] for $\mathcal{L}_{font}$ and $\mathcal{L}_{seg}$. Synthetic groundtruth is leveraged for fine-grained supervision and the task-oriented pre-training achieves fine-grained textural and spatial disentanglement of stylized text images which fertilizes the style representation for downstream generator.

3.3 Prior Guided Generation

With the robust glyph structure representation C_{struct} and fine-grained style disentanglement C_{style} mentioned above, a diffusion generator $\mathcal{G}$ is employed to integrate the prior guidance and generate the edited result as shown in Fig. 2 (c). For C_{struct}, since the U-Net in latent diffusion models contains both self-attention and cross-attention, wherein the cross-attention focuses on the relation between latent and external conditions [16, 17], we replace the key-value in cross-attention modules of $\mathcal{G}$ with the linear projection of C_{struct} to provide glyph guidance for improving accurate text rendering. For C_{style}, promising results have been shown by additional control injection [31] through the decoder of U-Net, based on which we apply the multi-scale style feature C_{style} to the skip-connections and middle block of the model $\mathcal{G}$ to provide a style reference for high-fidelity rendering.

With the leverage of pre-trained model [17], the training is performed under a combined supervision on the synthetic text image data. Please refer to Appendix A for preliminaries of the diffusion model and Appendix B.2 for implementation details of training.

3.4 Inference Control

During inference of the diffusion-based STE model, undesired color deviation and texture degradation occasionally occur. Such discrepancy can be partly attributed to the error accumulation during the iterative sampling process [27, 36]. Besides, the domain gap between training and inference impedes style consistency in complicated real-world scenery. To control the visual style consistency, we attempt to ameliorate the inference process by injecting style prior from the source image into editing. Specifically, we propose the Glyph-adaptive Mutual Self-Attention mechanism, which seamlessly incorporates the style of source images throughout the deconstruction process.

Reconstruction Branch. Rather than transforming random noise samples into an image, our objective is to execute an image-to-image translation, ensuring the preservation of style features. Initially, we perform DDIM inversion [27, 46] to generate an initial latent z^T_{source} from the source image I_{source}. The deconstructed inversion process enables a reconstruction branch $(z^T_{source}, z^{T-1}_{source}...z^0_{source})$ of the source image parallel to the editing branch $(z^T_{edit}, z^{T-1}_{edit}...z^0_{edit})$, which benefits the proposed integration process, as shown by the arrow in Fig. 2 (d).

Glyph-adaptive Mutual Self-Attention Mechanism (GaMuSa). Diverge from the general image editing, the target text modification of STE can lead to significant changes in the condition representation, which impedes text style preservation through general prompt-tuning methods [27].

Algorithm 1 Glyph-adaptive Mutual Self-attention

Input: Inversion latent z^T_{source}, reconstruction condition embedding c_{source}, editing condition embedding c_{edit} and target text embedding emb_y.
Parameters: Time step t, interval τ, intensity parameter λ and μ.
Output: Denoised latent z^0_{source} and z^0_{edit}.

```
$t = T, \lambda = 0, \mu = 1, \tau = 5$;
for $t = T, T-1...1$ do
    ———————— Reconstruction Branch ————————
    $z^{t-1}_{source}, \{K_s, V_s\} \leftarrow \hat{\mathcal{G}}(t, z^t_{source}, c_{source})$;    ▷ Self-Attention.
    ———————— Editing Branch ————————
    if at intervals of $\tau$ then
        $emb_{edit} = \mathcal{R}(\mathcal{E}_{dec}(z^t_{edit}))$;
        $\lambda = (emb_{edit} \cdot emb_y)/(\|emb_{edit}\| * \|emb_y\|)$;    ▷ Cosine Similarity.
    end if
    $\mu = 1 - \lambda$;
    $\{K_{es}, V_{es}\} = \lambda\{K_s, V_s\} + \mu\{K_e, V_e\}$;    ▷ Integration.
    $z^{t-1}_{edit} \leftarrow \hat{\mathcal{G}}(t, z^t_{edit}, c_{edit}; \{K_{es}, V_{es}\})$;    ▷ Glyph-adaptive Mutual Self-attention.
end for
Return $z^0_{source}, z^0_{edit}$
```

Recently, [36, 37] have demonstrated that self-attention layers in the diffusion model focus on latent internal relations. Based on the characteristic, we perform a mutual self-attention process between two branches as shown in Fig. 2(d), wherein at denoising step t, the Key-Value features $\{K_s, V_s\}$ from reconstruction branch are introduced to the self-attention operation in editing branch. Rather than a direct replacement in the previous method [36], however, we prefer an integration of $\{K_s, V_s\}$ and $\{K_e, V_e\}$ to mitigate the domain gap between z^t_{source} and z^t_{edit}.

It is worth noting that the original mutual self-attention process is hyper-parameter-sensitive to the starting time step. Premature initialization may introduce ambiguity and lead to spelling errors, whereas late intervention might fail to provide sufficient guidance. Inspired by the gradual deformation of text glyph during the iteration, we design a glyph-adaptive strategy to control the intensity of integration. Specifically, we employed a vision encoder $\mathcal{R}$ of the pre-trained text recognizer [47] to construct a glyph-adaptive strategy for the harmonious integration of Key-Value between branches. Specifically, during the iterative process, the intermediate latent z^t_{edit} is decoded and processed with $\mathcal{R}$ for the cosine similarity calculation with the target text embedding emb_y at intervals of τ steps to denote the glyph similarity of the intermediate edited image with target text. The similarity will serve as the intensity parameter λ and μ for controlling the integration between $\{K_s, V_s\}$ from reconstruction branch and $\{K_e, V_e\}$ from editing branch. The result of integration $\{K_{es}, V_{es}\}$ is subsequently leveraged in the self-attention modules of the editing branch for style coherency enhancement. The overall sampling pipeline is illustrated in Alg. 1.

$$\text{GaMuSa} = \text{Softmax}(\frac{Q_e \cdot (\lambda K_s + \mu K_e)^{\mathsf{T}}}{\sqrt{d}}) \cdot (\lambda V_s + \mu V_e), \quad \mu = 1 - \lambda. \tag{5}$$

4 Experiments

4.1 Dataset and Metrics

Training Data. Based on [1, 3], we synthesize 200k paired text images for style disentanglement pre-training and supervised training of TextCtrl, wherein each paired images are rendered with the same styles (i.e. font, size, colour, spatial transformation and background) and different texts, along with the corresponding segmentation mask and background image. Furthermore, a total of 730 fonts are employed to synthesize the visual text images in text glyph structure pre-training.

ScenePair Benchmark. To provide assessments on both visual quality and rendering accuracy, we propose the first real-world image-pair dataset in STE. Specifically, we collect 1,280 image pairs with

Methods \ Metrics	ScenePair (Cropped Text Image)				ScenePair (Full-size Image)	
	SSIM ↑ ($\times 10^{-2}$)	PSNR ↑	MSE ↓ ($\times 10^{-2}$)	FID ↓	SSIM ↑($\times 10^{-2}$)	FID ↓
SRNet [1]	26.66 ± 0.00	14.08 ± 0.00	5.61 ± 0.00	49.22 ± 0.00	98.91	1.48
MOSTEL [14]	27.45 ± 0.00	14.46 ± 0.00	5.19 ± 0.00	49.19 ± 0.00	98.96	1.49
DiffSTE [20]	26.85 ± 0.08	13.44 ± 0.04	6.11 ± 0.04	120.34 ± 1.52	98.86 (76.91)	2.37 (96.78)
TextDiffuser [21]	27.02 ± 0.11	13.96 ± 0.03	5.75 ± 0.05	57.01 ± 0.44	98.97 (92.76)	1.65 (12.23)
AnyText [22]	30.73 ± 0.55	13.66 ± 0.07	6.19 ± 0.14	51.79 ± 0.35	98.99 (82.57)	1.93 (16.92)
TextCtrl	**37.56** ± 0.32	**14.99** ± 0.15	**4.47** ± 0.15	**43.78** ± 0.17	**99.07**	**1.17**

Table 1: Text style fidelity assessment within text image level and full-size image level, highlighted with **best** and second best results. For full-size image evaluation, we replace the unedited region with the origin image while values in "()" denote the direct output of inpainting-based methods.

Methods \ Metrics	ScenePair		ScenePair (Random)		TamperScene [14]	
	ACC(%) ↑	NED ↑	ACC(%) ↑	NED ↑	ACC(%) ↑	NED ↑
SRNet [1]	17.84 ± 0.00	0.478 ± 0.000	9.61 ± 0.00	0.422 ± 0.000	39.96 ± 0.00	0.776 ± 0.000
MOSTEL [14]	37.69 ± 0.00	0.557 ± 0.000	22.50 ± 0.00	0.451 ± 0.000	**76.79** ± 0.00	0.858 ± 0.000
DiffSTE [20]	31.35 ± 0.35	0.538 ± 0.002	21.56 ± 0.69	0.487 ± 0.002	-	-
TextDiffuser [21]	51.48 ± 0.19	0.719 ± 0.003	33.99 ± 0.34	0.635 ± 0.004	-	-
AnyText [22]	51.12 ± 0.21	0.734 ± 0.005	25.05 ± 0.05	0.593 ± 0.003	-	-
TextCtrl	**84.67** ± 0.34	**0.936** ± 0.003	**66.95** ± 0.13	**0.869** ± 0.007	74.17 ± 0.55	**0.909** ± 0.011

Table 2: Text rendering accuracy evaluation with different methods, highlighted with **best** and second best results. "Random" denotes that we replace the paired target text in SCENEPAIR with randomly chosen text to verify the model robustness. Note that we are not able to evaluate inpainting-based STE methods [20, 21, 22] on TamperScene [14] since it does not contain full-size images.

text labels from ICDAR 2013 [48], HierText [49] and MLT 2017 [50], wherein each pair consists of two cropped text images with similar text length, style and background, along with the original full-size images. Collecting methods and dataset details are introduced in Appendix C.

Evaluation Dataset. For a fair comparison, we conduct all the evaluations on real-world datasets. ScenePair consists of 1,280 cropped text image pairs along with original full-size images enabling both style fidelity assessment and text rendering accuracy evaluation. TamperScene [14] combines a total of 7,725 cropped text images with predefined target text to provide rendering accuracy evaluation. Nevertheless, it does not involve paired images for style assessment nor full-size images for evaluation on inpainting-based methods, demonstrating the necessity of the proposed ScenePair.

Evaluation Metrics. For visual quality assessment, we adopt the commonly used metrics including (i) *SSIM*, mean structural similarity; (ii) *PSNR*, the ratio of peak signal to noise; (iii) *MSE*, the mean squared error on pixel-level; (iv) *FID* [51], the statistical difference between feature vectors. For text rendering accuracy comparison, we measure with (i) *ACC*, word accuracy and (ii) *NED*, normalized edit distance, using an official text recognition algorithm [52] and corresponding checkpoint.

4.2 Performance Comparison

Implementation. We conduct the comparison of the proposed TextCtrl with two GAN-based methods: SRNet [1] and MOSTEL [14] as well as three diffusion-based methods: DiffSTE [20], TextDiffuser [21] and AnyText [22] with their provided checkpoints. The quantitative results are illustrated in Tab. 1 and Tab. 2 while the qualitative results for comparison are shown in Fig. 3 and Fig. 4. Notably, DiffSTE [20], TextDiffuser [21] and AnyText [22] conduct STE with an inpainting manner on a full-size image, for which we employed the corresponding full-size image of each pair in ScenePair with the target text area masked as input and crop the generated target text area for style evaluation on text image level. SRNet [1], MOSTEL [14] and TextCtrl resolve STE in a synthesis manner on a text image, for which we perform a perspective process to paste the generated image back to the full-size image for style evaluation on full image level.

Text Style Fidelity. The text style fidelity assessment is performed on both the text image level and the full image level of ScenePair to enable a comprehensive comparison among methods. On

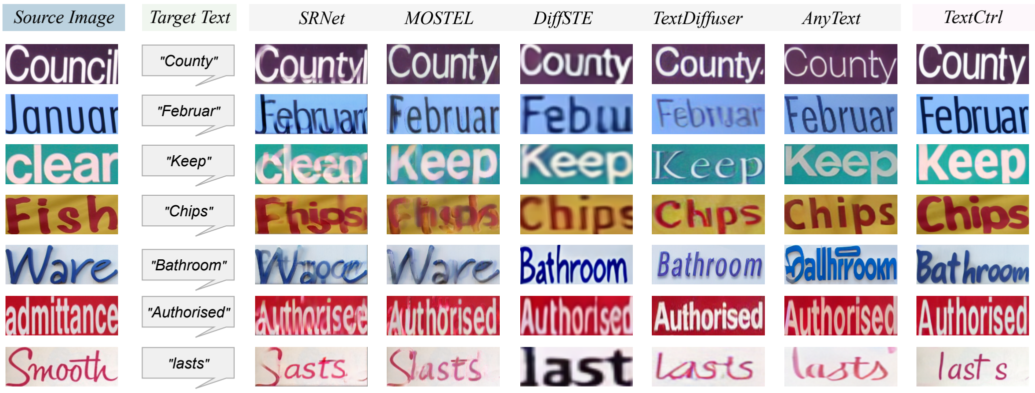


Figure 3: Qualitative comparison among different methods. Note that for the inpainting-based methods [20, 21, 22], we conduct the editing on the full-size images and perform the visualization of the edited text region.

text image level, TextCtrl outperforms other methods by at least 0.07, 0.53, 0.72 and 5.41 in SSIM, PSNR, MSE and FID respectively as represented in Tab. 1. GAN-based methods [1, 14] generally achieve a higher score on pixel-level assessment (i.e., PSNR, MSE). The reason lies in that they adopt a divide-and-conquer method, which contains a background restoration process to restrain the background region unaltered. Nevertheless, this may result in generating unsatisfied fuzzy images as shown in Fig. 3 column 3 and 4 due to the artifacts left by the unstable restoration process. Due to the loose style control result from the inpainting manner, undesired text style occurs occasionally in the outcome for diffusion-based methods [20, 21, 22]. On the contrary, TextCtrl benefits from the full leverage of disentangled style prior and the inference control for high-fidelity edited results. Further comparison on the full image level demonstrates the superiority of TextCtrl with precise manipulation and less style deviation against the inpainting-based methods with visualization in Fig. 4. Besides, it is not negligible that the inpainting strategy downgrades the image quality of unmasked regions.

Text Rendering Accuracy. Meanwhile, owing to the robust glyph structure representation, TextCtrl achieves superior spelling accuracy among all the methods, with more than 33% improvements in rendering accuracy of paired target text and randomly chosen text in ScenePair. TamperScene [14] contains a number of ambiguous low-resolution text images, which bring obstacles in style disentanglement and therefore impede the rendering accuracy of TextCtrl. Still, TextCtrl achieves a higher normalized edit distance that indicates the explicit mapping constructed between text and glyph. In comparison, GAN-based methods tend to yield ambiguous images, where source text left by imperfect removal blends with target text, resulting in fuzzy visual quality. Besides, due to the limited model capacity, GAN-based methods suffer from weak generalization and show incompetence with unseen style font as shown in Fig. 3 row 4 and 5. Diffusion-based methods achieve a disproportionate NED to their relatively low accuracy, which indicates their struggle with spelling mistakes. Notably, the inpainting manner serves as a primary factor that impedes the editing quality on small text for AnyText, whereas TextCtrl possesses the flexibility to perform editing on arbitrary scale text images.

4.3 Ablation Study

TextCtrl significantly improves STE through the substantial leverage of prior information in proposed (i) glyph structure representation pre-training, (ii) style disentanglement pre-training and (iii) glyph-adaptive mutual self-attention. We delve into the efficacy of each module in the following section.

Text Glyph Structure Representation. Conditional text prompt serves an important role in STE guiding the rendering of edited text glyphs. Consequently, we conduct the experiments by training TextCtrl with different text encoders $\mathcal{T}$ to evaluate the text rendering accuracy

Text Encoder	ScenePair		ScenePair (Random)	
	ACC(%) ↑	NED ↑	ACC(%) ↑	NED ↑
CLIP [40]	13.98	0.637	13.47	0.615
$\mathcal{T}$ w/o font-variance	76.08	0.875	60.84	0.827
$\mathcal{T}$ w font-variance	**84.67**	**0.936**	**66.95**	**0.869**

Table 3: Ablation experiment on glyph structure representation pre-training.

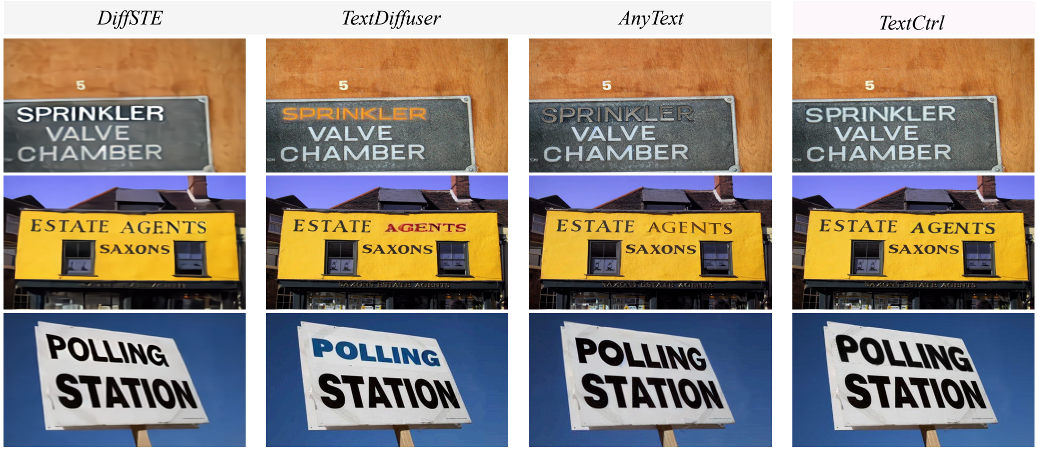


Figure 4: Qualitative comparison with inpainting-based methods [20, 21, 22] on full-size images.

on ScenePair. Specifically, we employed a CLIP text encoder [40] for comparison which is generally adopted in generative diffusion models [16, 17]. The contrast between ACC and NED results in Tab. 3 confirms that the CLIP text encoder [40] struggles with spelling mistakes which are attributed to the sub-word embedding [24] and sub-optimal alignment between text prompt and text glyph. We further analyze the impact of the proposed font-variance alignment strategy in pre-training and the results indicate the robust representation brought by the augmentation.

Text Style Disentanglement. The explicit text style disentanglement pre-training distinguishes TextCtrl from previous STE methods [14, 21, 22] in fostering the fine-grained feature representation ability on scene text concerning font, color, glyph and background texture. To further verify the style disentangling ability of TextCtrl, as depicted in Fig. 5, we visualize the style feature embedding using t-SNE [53] with text images from ICDAR 2013 [48] encoded by the pre-trained style encoder $\mathcal{S}$. From a broad perspective, text images with a similar color cluster in different regions of feature space which indicate the style representation of the image entirety. From a micro perspective, as shown by the sample pair pointed out with the magnifier, text images that share the same text style and background adjoin to each other regardless of different text content.

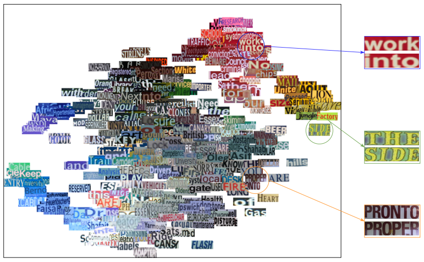


Figure 5: t-SNE [53] visualization of style features by pre-trained text style encoder.

In Tab. 4, we replace the style encoder with a prevalent module ControlNet following the implementation settings and empirical suggestions in [31]. Concretely, the style encoder is replaced with a vanilla Stable Diffusion encoder, serving as the ControlNet module. The module is initialized with the pre-trained weight of SD encoder. As a powerful technique in enhancing generation controllability, ControlNet is prevalently leveraged in image editing for style and structural control. The simple yet effective design enables a more meticulous reference from conditional input (e.g., Canny Edge, Depth map), which is also verified through the ablation study against our style encoder $\mathcal{S}$ (w/o pre-training) shown in Paper Tab. 4. After performing the style pre-training, however, the style encoder achieves a superiority performance against ControlNet module. Quantitative results demonstrate the fine-grained representation ability brought by the explicit style pre-training strategy, compared with implicit style learning by ControlNet. Notably, it also improves the parameter efficiency with 118M for style encoder $\mathcal{S}$ and 332M for ControlNet module.

Injection Module	SSIM ↑	MSE ↓	FID ↓
ControlNet [31]	0.3306	0.0464	58.30
$\mathcal{S}$ w/o pre-training	0.3130	0.0475	66.10
$\mathcal{S}$ w pre-training	**0.3756**	**0.0447**	**43.78**

Table 4: Ablation experiment on style disentanglement.

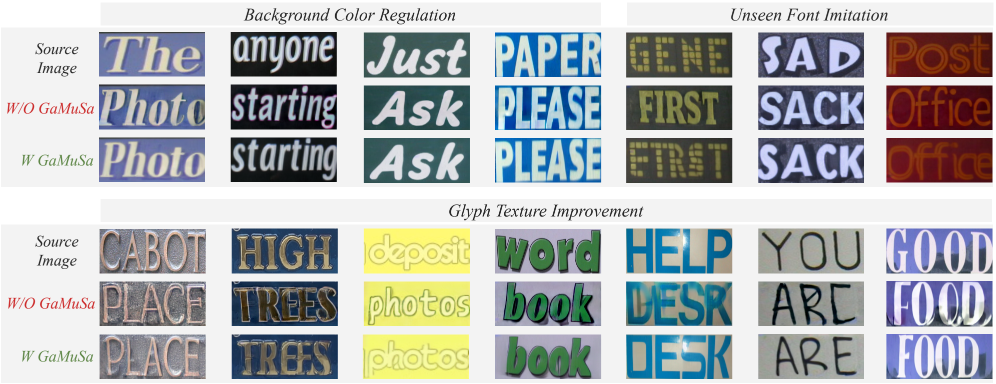


Figure 6: Visualization of text editing result with and without the proposed GaMuSa during inference, which verifies the improvements in background color, text font and glyph texture.

Inference control with Glyph-adaptive Mutual Self-attention (GaMuSa). In Tab. 5, we assess the style fidelity enhancement of GaMuSa in contrast with direct sampling and a prevalent enhancement method MasaCtrl [36] on ScenePair. Quantitative results verify the effectiveness of GaMuSa in enhancing style control during inference on variant real-world text images. Further visualization in Fig. 6 demonstrates the ability to persist style fidelity of GaMuSa when confronted with situations including background color deviation, unseen font and glyph texture degradation.

Inference	SSIM ↑	MSE ↓	FID ↓
w/o	0.3126	0.0609	51.35
w MasaCtrl [36]	0.3571	0.0468	49.53
w GaMuSa	**0.3756**	**0.0447**	**43.78**

Table 5: Ablation experiment on inference enhancement.

5 Limitations and Conclusion

Challenging arbitrary shape text editing. Arbitrary shape text editing occurs occasionally when editing with text on a crescent signboard or a circular icon as shown in Appendix Fig. 11. These texts possess a complicated geometric attribution which is hard to disentangle through the style reference. Early works [13, 14] adopt the Thin-Plate-Spline (TPS) module to capture the accurate geometric distribution of text and perform a transformation on the template image as pre-processing. However, this strategy only takes effect in GAN-based methods which adopt an image-to-image paradigm. It remains a problem to effectively introduce accurate geometric prior guidance to diffusion models.

Sub-optimal visual quality assessments metric. Following previous STE methods, we adopt a variety of evaluation metrics for visual quality assessment. However, these metrics either focus on pixel-level discrepancy or concentrate on feature similarity in latent space, which is sub-optimal for assessing text style coherency. Besides, all these metrics rely on the paired data under which a ground-truth image is required. Though we collect a real-world image-pair dataset ScenePair in our work, a large amount of real-world text images remain unpaired and thus fail to provide visual quality assessment in editing. While human evaluation may be a possible solution, a more efficient and objective visual assessment metric is expected for scene text editing.

In this paper, we propose a diffusion-based STE method named TextCtrl with the leverage of disentangled text style features and robust glyph structure guidance for high-fidelity text editing. For further coherency control during inference, a glyph-adaptive mutual self-attention mechanism is introduced along with the parallel sampling process. Additionally, an image-pair dataset termed ScenePair is collected to enable the comprehensive assessment on real-world images. Extensive quantitative experiments and qualitative results validate the superiority of TextCtrl.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant NO 62376266), and by the Key Research Program of Frontier Sciences, CAS (Grant NO ZDBS-LY-7024).

References

[1] Liang Wu, Chengquan Zhang, Jiaming Liu, Junyu Han, Jingtuo Liu, Errui Ding, and Xiang Bai. Editing Text in the Wild. In *Proceedings of the ACM International Conference on Multimedia*, 2019.

[2] Yan Shu, Weichao Zeng, Zhenhang Li, Fangmin Zhao, and Yu Zhou. Visual Text Meets Low-level Vision: A Comprehensive Survey on Visual Text Processing. *arXiv preprint arXiv:2402.03082*, 2024.

[3] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic Data for Text Localisation in Natural Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2315–2324, 2016.

[4] Praveen Krishnan, Rama Kovvuri, Guan Pang, Boris Vassilev, and Tal Hassner. TextStyleBrush: Transfer of Text Aesthetics From a Single Example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:9122–9134, 2021.

[5] Victor Fragoso, Steffen Gauglitz, Shane Zamora, Jim Kleban, and Matthew A. Turk. TranslatAR: A Mobile Augmented Reality Translator. In *Proceedings of the IEEE Workshop on Applications of Computer Vision*, pages 497–502, 2011.

[6] Yan Shu, Wei Wang, Yu Zhou, Shaohui Liu, Aoting Zhang, Dongbao Yang, and Weipinng Wang. Perceiving Ambiguity and Semantics without Recognition: An Efficient and Effective Ambiguous Scene Text Detector. In *Proceedings of the ACM International Conference on Multimedia*, pages 1851–1862, 2023.

[7] Zhi Qiao, Yu Zhou, Dongbao Yang, Yucan Zhou, and Weiping Wang. SEED: Semantics Enhanced Encoder-decoder Framework for Scene Text Recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13528–13537, 2020.

[8] Zhi Qiao, Yu Zhou, Jin Wei, Wei Wang, Yuan Zhang, Ning Jiang, Hongbin Wang, and Weiping Wang. PIMNet: A Parallel, Iterative and Mimicking Network for Scene Text Recognition. In *Proceedings of the ACM International Conference on Multimedia*, pages 2046–2055, 2021.

[9] Wei Wang, Yu Zhou, Jiahao Lv, Dayan Wu, Guoqing Zhao, Ning Jiang, and Weipinng Wang. TPSNet: Reverse Thinking of Thin Plate Splines for Arbitrary Shape Scene Text Representation. In *Proceedings of the ACM International Conference on Multimedia*, pages 5014–5025, 2022.

[10] Huawen Shen, Xiang Gao, Jin Wei, Liang Qiao, Yu Zhou, Qiang Li, and Zhanzhan Cheng. Divide Rows and Conquer Cells: Towards Structure Recognition for Large Tables. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1369–1377, 2023.

[11] Gangyan Zeng, Yuan Zhang, Yu Zhou, Xiaomeng Yang, Ning Jiang, Guoqing Zhao, Weiping Wang, and Xu-Cheng Yin. Beyond OCR + VQA: Towards End-to-end Reading and Reasoning for Robust and Accurate TextVQA. *Pattern Recognition*, 138:109337, 2023.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. *Advances in Neural Information Processing Systems*, 27, 2014.

[13] Qiangpeng Yang, Hongsheng Jin, Jun Huang, and Wei Lin. SwapText: Image Based Texts Transfer in Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14688–14697, 2020.

[14] Yadong Qu, Qingfeng Tan, Hongtao Xie, Jianjun Xu, Yuxin Wang, and Yongdong Zhang. Exploring Stroke-Level Modifications for Scene Text Editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2119–2127, 2023.

[15] Prafulla Dhariwal and Alex Nichol. Diffusion Models Beat GANs on Image Synthesis. *arXiv preprint arXiv:2105.05233*, 2021.

[16] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv preprint arXiv:2204.06125*, 2022.

[17] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10674–10685, 2022.

[18] Changshuo Wang, L. Wu, Xu Chen, Xiang Li, Lei Meng, and Xiangxu Meng. Letter Embedding Guidance Diffusion Model for Scene Text Editing. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 588–593, 2023.

[19] Joshua Santoso, Christian Simon, et al. On Manipulating Scene Text in the Wild with Diffusion Models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5202–5211, 2024.

[20] Jiabao Ji, Guanhua Zhang, Zhaowen Wang, Bairu Hou, Zhifei Zhang, Brian L Price, and Shiyu Chang. Improving Diffusion Models for Scene Text Editing with Dual Encoders. *Transactions on Machine Learning Research*, 2023.

https://doi.org/10.52202/079017-4396

[21] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. TextDiffuser: Diffusion Models as Text Painters. *Advances in Neural Information Processing Systems*, 36, 2024.

[22] Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Geng, and Xuansong Xie. AnyText: Multilingual Visual Text Generation And Editing. In *Proceedings of the International Conference on Learning Representations*, 2024.

[23] Haoxing Chen, Zhuoer Xu, Zhangxuan Gu, Yaohui Li, Changhua Meng, Huijia Zhu, Weiqiang Wang, et al. DiffUTE: Universal Text Editing Diffusion Model. *Advances in Neural Information Processing Systems*, 36, 2024.

[24] Rosanne Liu, Daniel H Garrette, Chitwan Saharia, William Chan, Adam Roberts, Sharan Narang, Irina Blok, R. J. Mical, Mohammad Norouzi, and Noah Constant. Character-Aware Models Improve Visual Text Rendering. In *Proceedings of the Association for Computational Linguistics*, 2022.

[25] Yukang Yang, Dongnan Gui, Yuhui Yuan, Weicong Liang, Haisong Ding, Han Hu, and Kai Chen. GlyphControl: Glyph Conditional Control for Visual Text Generation. *Advances in Neural Information Processing Systems*, 36, 2024.

[26] Zhenhang Li, Yan Shu, Weichao Zeng, Dongbao Yang, and Yu Zhou. First Creating Backgrounds Then Rendering Texts: A New Paradigm for Visual Text Blending. In *Proceedings of the European Conference on Artificial Intelligence*, 2024.

[27] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text Inversion for Editing Real Images using Guided Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023.

[28] Jeyasri Subramanian, Varnith Chordia, Eugene Bart, Shaobo Fang, Kelly Guan, Raja Bala, et al. STRIVE: Scene Text Replacement In Videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14549–14558, 2021.

[29] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8107–8116, 2019.

[30] Yiming Zhao and Zhouhui Lian. UDiffText: A Unified Framework for High-quality Text Synthesis in Arbitrary Images via Character-aware Diffusion Models. *arXiv preprint arXiv:2312.04884*, 2023.

[31] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3813–3824, 2023.

[32] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023.

[33] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2022.

[34] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *Proceedings of the International Conference on Learning Representations*, 2021.

[35] Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. In *Proceedings of the NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[36] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22560–22570, 2023.

[37] Yuval Alaluf, Daniel Garibi, Or Patashnik, Hadar Averbuch-Elor, and Daniel Cohen-Or. Cross-Image Attention for Zero-Shot Appearance Transfer. *arXiv preprint arXiv:2311.03335*, 2023.

[38] Shuyuan Tu, Qi Dai, Zhi-Qi Cheng, Han Hu, Xintong Han, Zuxuan Wu, and Yu-Gang Jiang. MotionEditor: Editing Video Motion via Content-Aware Diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7882–7891, 2024.

[39] Haibin He, Xinyuan Chen, Chaoyue Wang, Juhua Liu, Bo Du, Dacheng Tao, and Qiao Yu. Diff-Font: Diffusion Model for Robust One-shot Font Generation. *International Journal of Computer Vision*, pages 1–15, 2024.

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning*, 2021.

[41] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion. In *Proceedings of the International Conference on Learning Representations*, 2022.

[42] Xun Huang and Serge J. Belongie. Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1510–1519, 2017.

[43] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid Scene Parsing Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6230–6239, 2016.

[44] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In-So Kweon. CBAM: Convolutional Block Attention Module. *arXiv preprint arXiv:1807.06521*, 2018.

[45] Fausto Milletarì, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In *Proceedings of the International Conference on 3D Vision*, pages 565–571, 2016.

[46] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. DiffEdit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022.

[47] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. Read Like Humans: Autonomous, Bidirectional and Iterative Language Modeling for Scene Text Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7094–7103, 2021.

[48] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, M. Iwamura, Lluís Gómez i Bigorda, Sergi Robles Mestre, Joan Mas Romeu, David Fernández Mota, Jon Almazán, and Lluís-Pere de las Heras. ICDAR 2013 Robust Reading Competition. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 1484–1493, 2013.

[49] Shangbang Long, Siyang Qin, Dmitry Panteleev, Alessandro Bissacco, Yasuhisa Fujii, and Michalis Raptis. ICDAR 2023 Competition on Hierarchical Text Detection and Recognition. *arXiv preprint arXiv:2305.09750*, 2023.

[50] Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, Yuan Feng, Dimosthenis Karatzas, Zhenbo Luo, Umapada Pal, Christophe Rigaud, Joseph Chazalon, Wafa Khlif, Muhammad Muzzamil Luqman, Jean-Christophe Burie, Cheng-Lin Liu, and Jean-Marc Ogier. ICDAR2017 Robust Reading Challenge on Multi-Lingual Scene Text Detection and Script Identification - RRC-MLT. In *Proceedings of the International Conference on Document Analysis and Recognition*, volume 01, pages 1454–1459, 2017.

[51] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017.

[52] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4714–4722, 2019.

[53] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[54] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[55] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[56] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015.

[57] Rowel Atienza. Vision Transformer for Fast and Efficient Scene Text Recognition. In *Proceedings of the IEEE International Conference on Document Analysis and Recognition*, 2021.

[58] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929*, 2020.

[59] K Simonyan and A Zisserman. Very Deep Convolutional Networks for Large-scale Image Recognition. In *Proceedings of the International Conference on Learning Representations*, 2015.

https://doi.org/10.52202/079017-4396

Appendix

A Preliminaries

Latent Diffusion Models (LDMs). Instead of operating diffusion process [34, 54] in image pixel space, LDMs utilize an autoencoder [55] ε to translate the input image x to latent space representation z_0. Then the denoising network ϵ_θ built upon a time-conditional UNet [56] is trained to estimate added noise ϵ at a time step t. The condition embedding c, which is $c = \{C_{struct}, C_{style}\}$ in our work, is integrated through the cross-attention mechanism or the skip-connections and middle block, realizing a conditional generation that has the following training objective:

$$\mathcal{L}_{dn} = \mathbb{E}_{\varepsilon(x),c,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(t, z_t, c)\|_2^2\right], \tag{6}$$

DDIM Sampling. At inference, random Gaussian noised z_T can be gradually denoised to form a result z_0 through iterative sampling, wherein the deterministic DDIM sampling [34] is adopted in our method:

$$z_{t-1} = \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} z_t + \sqrt{\alpha_{t-1}}(\sqrt{\frac{1}{\alpha_{t-1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1}) \cdot \epsilon_\theta(t, z_t, c), \tag{7}$$

where $\alpha_t = \prod_{i=1}^{t}(1 - \beta_i)$ and $\beta_0 = 0$ and tends to 1 as i increases.

DDIM Inversion. In contrast to the stochastic sampling employed in DDPM [54], the sampling process in DDIM [34] is deterministic, allowing for the complete inversion [27, 46] from the original images latent z_0 back to initial noised latent z_T to construct the reconstruction branch in our work.

$$z_{t+1} = \sqrt{\frac{\alpha_{t+1}}{\alpha_t}} z_t + \sqrt{\alpha_{t+1}}(\sqrt{\frac{1}{\alpha_{t+1}} - 1} - \sqrt{\frac{1}{\alpha_t} - 1}) \cdot \epsilon_\theta(t, z_t, c), \tag{8}$$

Classifier-Free Guidance (CFG). To improve the visual quality and faithfulness of generated images, [35] introduces the classifier-free guidance technique, which jointly trains a conditional and an unconditional (denoted as c_{null}) diffusion model to provide a refined result:

$$\hat{\epsilon}_\theta(t, z_t, c) = \omega \cdot \epsilon_\theta(t, z_t, c) + (1 - \omega) \cdot \epsilon_\theta(t, z_t, c_{null}), \tag{9}$$

where ω is a hyperparameter that controls the strength of guidance.

B Implementation Setting

B.1 Details of Model Architecture and Parameter

TextCtrl primarily comprises five components: an Encoder-Decoder VAE $\mathcal{E}$, a U-Net backbone $\mathcal{G}$, a text glyph structure encoder $\mathcal{T}$, a text style encoder $\mathcal{S}$ and a vision encoder $\mathcal{R}$. For the VAE and U-Net, we employ the pre-trained checkpoint of Stable Diffusion [17] V1-5[2]. For the text glyph structure encoder, we utilize a lightweight transformer encoder and perform pre-training on the proposed glyph structure representation aligning to the visual features captured by a frozen vision encoder[3] [57]. For the text style encoder, a ViT [58] backbone is employed to perform pre-training on multi-task style disentanglement. For the vision encoder, the vision backbone of ABINet[4] [47] is adopted with pre-trained checkpoint.

As the model input, the source image is resized to 256 × 256 while the max target prompt length is set to 24. The training process utilizes a batch size of 256 with a learning rate of 1×10^{-5} and a total epoch of 100. TextCtrl is trained on 4 NVIDIA A6000 GPU and the parameter sizes of each module are provided in Tab. 6.

[2] https://huggingface.co/runwayml/stable-diffusion-v1-5

[3] https://github.com/roatienza/deep-text-recognition-benchmark

[4] https://github.com/FangShancheng/ABINet

Modules	$\mathcal{G}$	$\mathcal{E}$	$\mathcal{T}$	$\mathcal{S}$	$\mathcal{R}$	Total
Params	859M	83M	66M	118M	90M	1216M

Table 6: The parameter sizes of each module in TextCtrl

B.2 Details of Training

During training of the generator, we follow the Diffusion Denoising Probabilistic Models (DDPM) [54] and perform the forward noising process on the image latent $z_0^e = \mathcal{E}_{enc}(I_e)$ with random $t \in \{1, ..., T\}$ to generate noised latent z_t^e:

$$z_t^e = \sqrt{\alpha_t} z_0^e + \sqrt{1-\alpha_t}\epsilon_t, \epsilon_t \sim \mathcal{N}(0, I), \tag{10}$$

where $\alpha_t = \prod_{i=1}^{t}(1-\beta_t)$ and $\beta_t \in (0,1)$ is defined by a parameter schedule. The noise latent serves as the model input yielding the predicted noise $\tilde{\epsilon}_t$ along with condition embedding $c = \{C_{struct}, C_{style}\}$ as:

$$\tilde{\epsilon}_t = \mathcal{G}(t, z_t^e, c), \tag{11}$$

during which c is randomly replaced by the unconditional embedding $c_{null} = \{C_{null}, C_{style}\}$ with the probability $p_{uc} = 0.1$ to jointly train an unconditional model, enabling the leverage of classifier-free guidance [35].

Since the diffusion process is performed in the latent space, to enable further vision and linguistics supervision, we construct the image-level result $\tilde{I}_e = \mathcal{E}_{dec}(\tilde{z}_e^0)$ where $\tilde{z}_e^0$ is gained through:

$$\tilde{z}_0^e = \frac{1}{\sqrt{\alpha_t}}(z_t^e - \sqrt{1-\alpha_t}\tilde{\epsilon}_t), \tag{12}$$

We provide a triple-guidance loss for supervision of the diffusion generator, including the denoising loss $\mathcal{L}_{dn}$, the construction loss $\mathcal{L}_{cons}$ and the linguistic loss $\mathcal{L}_{ocr}$.

For construction loss, we adopt the $\mathcal{L}_{per}$ and $\mathcal{L}_{style}$ as part of construction loss following [14], along with MSE loss $\mathcal{L}_{reg}$. The functions are written as:

$$\mathcal{L}_{cons} = \lambda_1 \mathcal{L}_{per} + \lambda_2 \mathcal{L}_{style} + \mathcal{L}_{reg}, \tag{13}$$

$$\mathcal{L}_{per} = \mathbb{E}[\|\phi_i(I_e) - \phi_i(\tilde{I}_e)\|_1], \tag{14}$$

$$\mathcal{L}_{style} = \mathbb{E}_j[\|G_j^{\phi}(I_e) - G_j^{\phi}(\tilde{I}_e)\|_1], \tag{15}$$

where balance factors λ_1 and λ_2 are set to 0.01 and 100 respectively. ϕ_i is the activation map from *relu1_1* and *relu5_1* layer of VGG-19 model [59] and G is the Gram matrix.

Words are composed of character sequences that inherently contain linguistic information. pre-trained text recognition models, rich in sequence features prior, can be harnessed as global guidance to enhance text rendering ability. For linguistic loss, a recognition process will be applied to the decoded image $\tilde{I}_e \in \mathbb{R}^{3\times H\times W}$ generating recognition result $\tilde{y}$, which is utilized to calculate the cross-entropy loss CE with the text label y:

$$\mathcal{L}_{ocr} = \lambda_3 CE(y, \tilde{y}), \tag{16}$$

where λ_3 is set to 0.01. The overall objective function in training can be presented as a combination of the denoising loss Eq. 6, the construction loss Eq. 13 and the text recognition loss Eq. 16:

$$\mathcal{L} = \mathcal{L}_{dn} + \mathcal{L}_{cons} + \mathcal{L}_{ocr}. \tag{17}$$

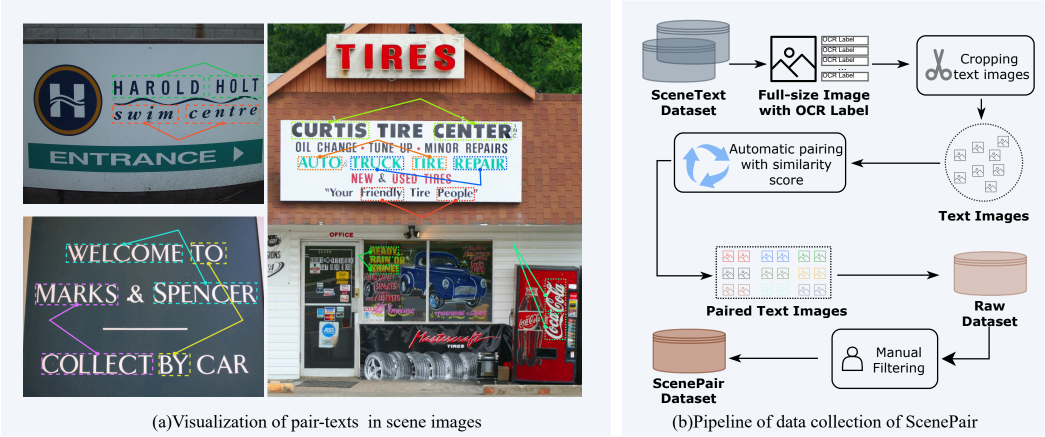
B.3 Details of Inference

During inference, for a full-size image, the quadrangle location of the source text region is indicated through user interface or detection with ocr model, which is the same as all the other STE methods to provide clear instruction of the specific text to be edited. Subsequently, we perform a cropping and perspective process on the text region to acquire the input source image. The source image is further employed in an inversion process based on Eq. 8 to construct a reconstruction branch and disentangled in style encoder $\mathcal{S}$ to serve as style guidance C_{style}, while the arbitrary target text provided by user is encoder by glyph structure encoder $\mathcal{T}$ as C_{struct}. The condition embedding

https://doi.org/10.52202/079017-4396

$c = \{C_{style}, C_{strcut}\}$ is later passed to the generator $\mathcal{G}$ guiding the generation process. The sampling step is set to $T = 50$ and the classifier-free guidance scale is set to $\omega = 2$ with 7 seconds to generate an edited image on a single NVIDIA A6000 GPU. Note that for the reconstruction branch, either a default null text or a source text is suitable for the input of glyph structure encoder $\mathcal{T}$ due to the symmetry of the inversion process Eq. 8 and the reconstruction process Eq. 7, which is flexible and will not interface the editing quality. The edited result will be perspective and stitched back to the original region for the full image result.

C ScenePair Dataset



(a)Visualization of pair-texts in scene images

(b)Pipeline of data collection of ScenePair

Figure 7: Data collection strategy of ScenePair. (a) Texts with the same style and background often occur in real-world scenery. (b) The pipeline for our data collection.

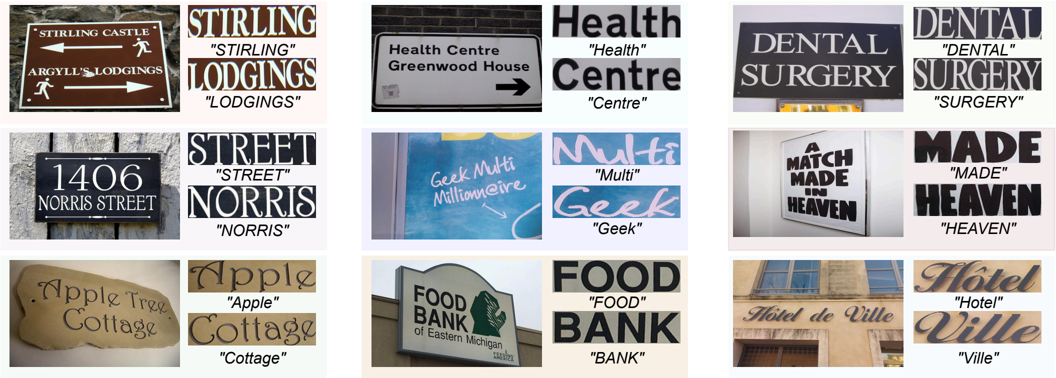


Figure 8: Visualization of paired data along with full-size images in SCENEPAIR.

Dataset Introduction. To provide practical assessments on both style consistency and rendering accuracy, we propose the first real-world image-pair dataset in STE termed ScenePair. Specifically, we collect 1,280 image pairs with the text label from ICDAR 2013 [48], HierText [49] and MLT 2017 [50]. For each pair, we collect the source text image, the target text image, the respective text labels, the respective quadrangle locations in full-size image and the original full-size image.

The inspiration for constructing ScenePair dataset comes from the observation that scene texts often occur in phrases with the same style and background in real-world scenery, as depicted in Fig. 7 (a), which serves as a perfect pair sample for evaluation of editing quality. Notably, there isn't a "correct" result for image editing whereas our paired data serves as a reference benchmark for high-fidelity.

Collecting Strategy. We design a semi-automatic collecting strategy for ScenePair from several scene text datasets as illustrated in Fig. 7 (b). Initially, we collect the full-size images from the

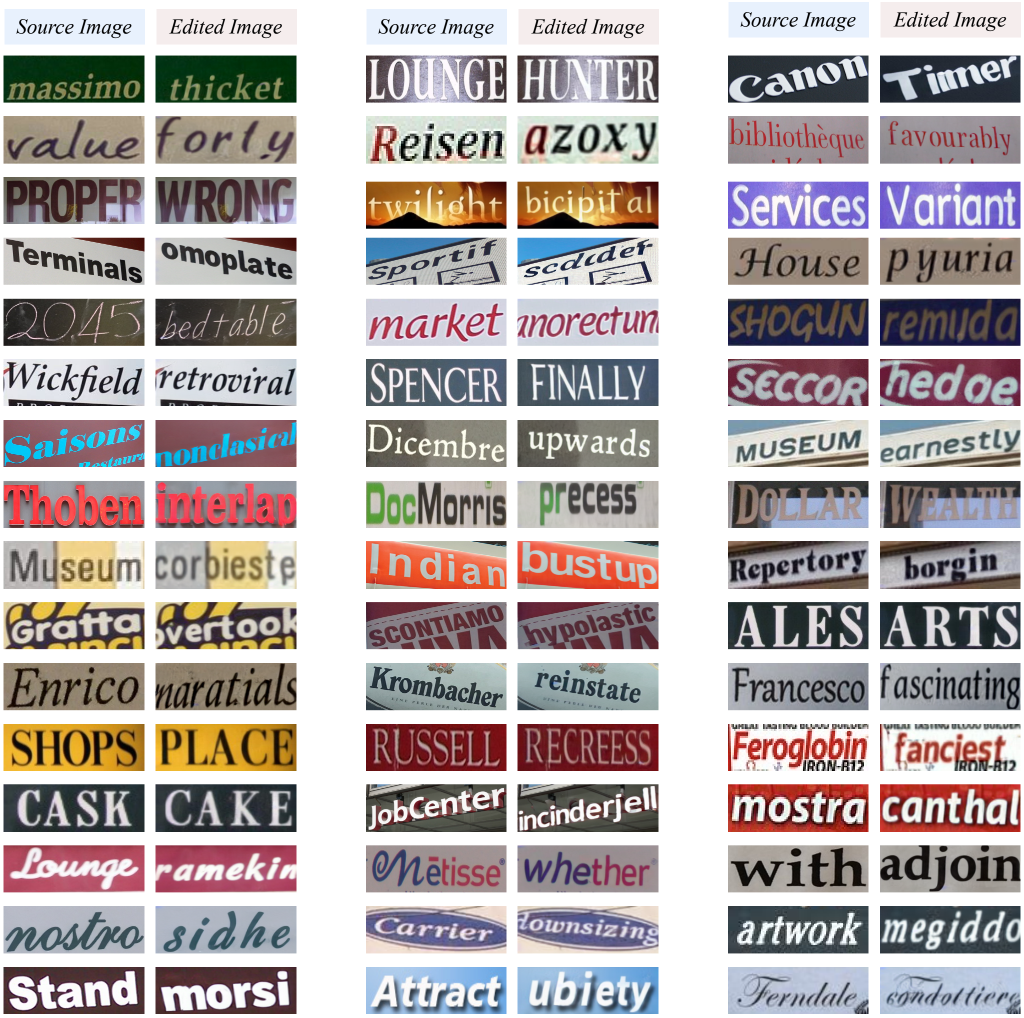


Figure 9: Visualization of edited result on text image by TEXTCTRL on dataset TAMPERSCENE.

datasets along with the detection and recognition labels. For each full-size image, we perform cropping and perspective with the quadrangle location to acquire all the text images, for which an automatic pairing algorithm is employed to construct paired text images. Specifically, the algorithm calculates the weighted similarity score involving text length, aspect ratio, centre distance in full-size image and SSIM (Structure Similarity Index Measure) between the cropped text images, wherein the paired images with similarity score that surpass a pre-defined threshold would be chosen to form a raw dataset. Finally, we manually filter out the unsatisfied pairs to construct the ScenePair dataset, as shown in Fig. 8.

D Visualization

To further demonstrate the editing performance of our TextCtrl, we provide various visualizations of edited outcomes. In Fig. 9, a variety of images in TamperScene with abundant text style are provided to verify the text style generalization of TextCtrl. In Fig. 10, we provide the edited result on the scene image of ICDAR 2013. Specifically, given a scene photo with an automatically detected text box, we perform scene text editing on box images and stitch back to the original photo. Results demonstrate the ability of TextCtrl to preserve fine-grained detail for high-fidelity editing.

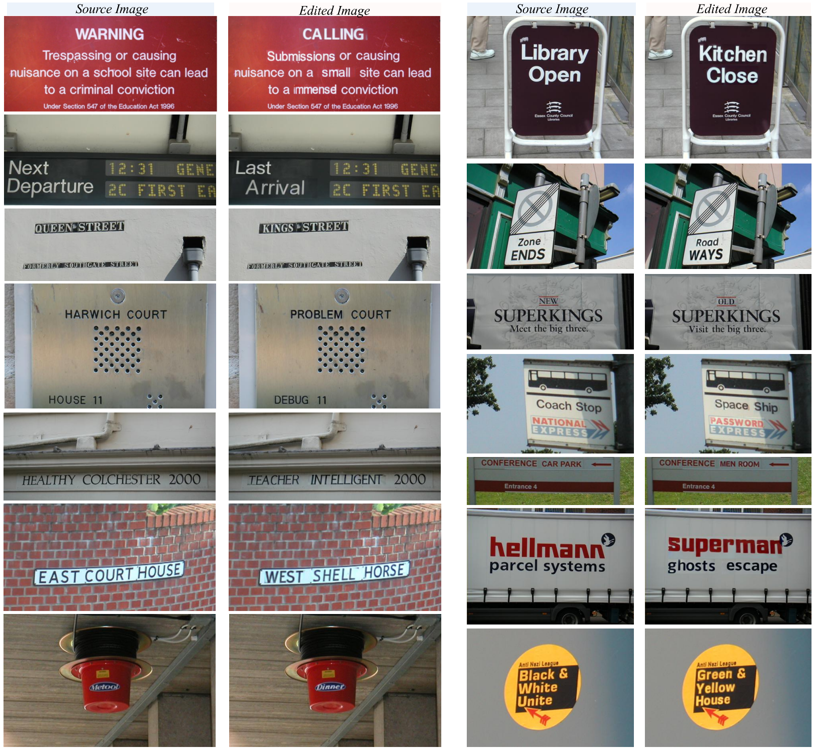


Figure 10: Visualization of edited result on scene image by TEXTCTRL on dataset ICDAR 2013.

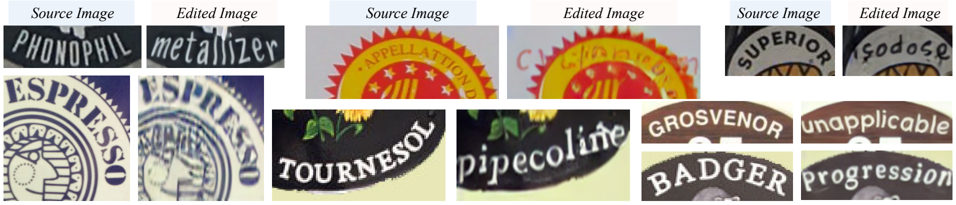


Figure 11: Visualization of failure cases on text image by TEXTCTRL. The problem mainly results from the insufficient geometric prior guidance control.

E Additional Considerations

This paper introduces a novel method for scene text editing which leverages the fine-grained style disentanglement and robust glyph structure representation to achieve high-fidelity editing results. Though we acknowledge that the proposed method has the potential to be misused for image forgery, significant advancement in visual quality and text rendering accuracy would also contribute to text-related visual art creation. To prevent the high risk of misuse of the proposed method, an additional user commitment will be required for accessing the checkpoint in our forthcoming open release, through which we hope to alleviate the potential misuse while benefiting further research.

F Licenses

Here we provide license details of the code and data used in our proposed network and comparison experiments. SRNet [1] is available for use under GNU General Public License v3.0. AnyText [22], deep-text-recognition-benchmark [52], ViTSTR [57] are available for use under Apache License 2.0. DiffSTE [20], TextDiffuser [21], CLIP [40] are available for use under MIT License. Stable Diffusion [16] is available for research purposes under CreativeML Open RAIL M License. ABINet

https://doi.org/10.52202/079017-4396

is available for non-commercial purposes with license [5]. MLT 2017 [50] and HierText [49] are released under CC BY-SA 4.0 license. There is no known license for ICDAR 2013[6] [48], MOSTEL[7] [14] and TamperScene[8] [14], but the data and code are commonly referred to as "public", and so we interpret this to mean they are available for use under research purpose.

[5] https://github.com/FangShancheng/ABINet/blob/main/LICENSE

[6] https://rrc.cvc.uab.es/?ch=2

[7] https://github.com/qqqyd/MOSTEL

[8] https://github.com/qqqyd/MOSTEL

https://doi.org/10.52202/079017-4396

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly reflect the main contributions and scope of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations of our work. Please refer to Sec. 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

https://doi.org/10.52202/079017-4396

Justification: Our propositions are mainly based on experiments and empirical results which do not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided all the implementation details of model design, network training and inference in Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

https://doi.org/10.52202/079017-4396

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Project page: https://github.com/weichaozeng/TextCtrl.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We have provided all the details including the data and hyperparameter in our paper. Please refer to Sec. 4.1 and Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report the standard deviation with respect to the random seed after running each model 3 times in our main experiments. Please refer to Tab. 1 and Tab. 2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

https://doi.org/10.52202/079017-4396

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided these details in the paper. Please refer to Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conducted in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed the potential societal impacts of our work. Please refer to Appendix E.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

https://doi.org/10.52202/079017-4396

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We describe the safeguards that we will put in place in our future release of checkpoints. Please refer to Appendix E.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited all relevant existing works and assets which are related/used in our work in References. We also provide license details of the assets used in our work. Please refer to Appendix F.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Project page: https://github.com/weichaozeng/TextCtrl.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

https://doi.org/10.52202/079017-4396

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.