

# StyleTextGen: Style-Conditioned Multilingual Scene Text Generation

Zeyu Chen<sup>1</sup> Fangmin Zhao<sup>3</sup> Yan Shu<sup>2</sup> Yichao Liu<sup>1</sup> Liu Yu<sup>1</sup> Yu Zhou<sup>1\*</sup>  
<sup>1</sup>Nankai University <sup>2</sup>University of Trento  
<sup>3</sup>Institute of Information Engineering, Chinese Academy of Sciences

## Abstract

Style-conditioned scene text generation faces unique challenges in extracting precise text styles from complex backgrounds and maintaining fine-grained style consistency across characters, especially for multilingual scripts. We propose StyleTextGen, a novel framework that learns to perceive and replicate visual text styles across different languages and writing systems. Our approach features three key contributions: First, we introduce a dual-branch style encoder dedicated to style modeling, yielding robust multilingual text style representations in complex real-world scenes. Second, we design a text style consistency loss that enhances style coherence and improves overall visual quality. Third, we develop a mask-guided inference strategy that ensures precise style alignment between generated and reference text. To facilitate systematic evaluation, we construct StyleText-CE, a bilingual scene text style benchmark covering both monolingual and cross-lingual settings. Extensive experiments demonstrate that StyleTextGen significantly outperforms existing methods in style consistency and cross-lingual generalization, establishing new state-of-the-art performance in multilingual style-conditioned text generation.

## 1. Introduction

Text-to-image generation [20, 32] has witnessed remarkable progress in synthesizing photorealistic images from textual descriptions. However, generating scene text with precise style control remains a fundamental challenge. The task of style-conditioned scene text generation aims to synthesize text in images while faithfully replicating the visual style from reference example, including font characteristics, stroke pattern, color gradient and texture detail. This capability is crucial for real-world applications such as multilingual signage design and advertising poster creation, and also benefit scene text detection, recognition, and spotting [2, 7, 8, 28, 51, 60]. Unlike semantic content that

\*Corresponding author.



Figure 1. Examples of our StyleTextGen for style-conditioned scene text generation. Left: self-style reference (in-place generation). Right: external-style reference. For each example, the reference style image and the generated target text are shown at the bottom of the panel.

can be specified through text prompts, these nuanced visual attributes require visual style references, making image-conditioned generation a natural solution for this task.

Despite its importance, style-conditioned scene text generation faces unique technical challenges that distinguish it from general image synthesis. First, extracting text-specific style features from complex scene images is

non-trivial—reference images often contain cluttered backgrounds, varying lighting conditions, and diverse text layouts that can confound style perception. Existing scene text editing methods [9, 56] typically reuse styles from the original in-place text, limiting their ability to adopt external style references. Recent advances such as Calligrapher [30] attempt free-style text generation, but their style extraction mechanisms lack explicit content awareness, leading to style-content entanglement and degraded performance when processing complex backgrounds. Second, maintaining fine-grained style consistency across all characters becomes substantially more challenging in multilingual scenarios, where different writing systems (e.g., Latin, Chinese, Arabic) exhibit distinct typographic structures and varying numbers of strokes. The lack of robust cross-lingual style transfer mechanisms in current methods results in inconsistent character appearance and structural distortions when generating text in languages different from the reference style.

To address these challenges, we propose StyleTextGen, a novel style-conditioned framework for multilingual scene text generation. First, we design a dual-branch style encoder dedicated to modeling text style. A text-focused branch pretrained on multilingual text segmentation captures fine-grained typographic features, while a visual prior branch extracts holistic scene information. Their fusion yields robust, text-aware style embeddings that remain stable across complex backgrounds and diverse languages. Second, we propose a text style consistency loss that enforces style alignment specifically within text regions. This mask-guided loss promotes uniform style attributes across all generated characters, significantly improving visual coherence in multilingual scenarios. Third, we introduce a mask-guided inference strategy that extracts style features from reference images through mask-based inversion and injects them into the generation process. This refinement ensures precise style correspondence between reference and generated text across different writing systems.

To enable systematic evaluation, we construct StyleTextCE, a bilingual benchmark supporting four settings: self-style reference, external-style reference, monolingual generation, and cross-lingual transfer. As shown in Figure 1, StyleTextGen accurately transfers text styles while maintaining readability across diverse languages and styles.

Our contributions are summarized as follows:

- We propose a dual-branch style encoder that robustly captures style features from multilingual text, combined with a text style consistency loss to enhance style coherence and visual consistency.
- We design an inference strategy that enhances style alignment between generated scene text and reference, improving fine-grained style consistency across characters.
- We construct a bilingual scene text style benchmark

with monolingual and cross-lingual settings, demonstrating superior performance in style consistency and cross-lingual generalization.

## 2. Related Work

### 2.1. Scene Text Generation

Traditional text-to-image diffusion models [31, 35, 36, 39–41, 53], struggle to render legible and semantically correct text in complex scenes. This limitation has motivated a line of work on dedicated scene text generation. Methods such as DiffSTE [13], UDiffText [59], GlyphByT5 [26], and SceneTextGen [58] improve textual representation through character-level encoders. Apart from these, AnyText [43], GlyphDraw2 [29], GlyphOnly [24], TextDiffuser [5], TextDiffuser2 [4], and GlyphControl [52] leverage glyph priors and layout masks to enhance structural fidelity and spatial controllability.

Focusing more specifically on style-conditioned scene text generation, GlyphByT5 [26] models partial font and color information within its text encoder and pre-rendered glyph space, enabling coarse-grained control over the appearance of generated text. AnyText2 [44] further introduces explicit font and color encoders to provide additional style-related guidance during rendering. However, these approaches are typically restricted to relatively simple fonts and often struggle to synthesize artistic typography or faithfully follow a given style reference. To address these limitations, we propose StyleTextGen, which generates scene text conditioned on arbitrary text-style reference images.

### 2.2. Scene Text Editing

Scene text editing aims to modify the textual content in an image while preserving the background and maintaining style consistency. Many methods explicitly separate foreground text from background. SRNet [48] and STEFANN [38] use dedicated modules for background inpainting and text re-rendering. SwapText [50] extends SRNet with a thin-plate spline interpolation network to handle curved text. TextStyleBrush [18] adopts a StyleGAN2 [17] based self-supervised framework to learn transferable text styles, and MOSTEL [33] employs a semi-supervised training scheme to improve robustness in the wild. STEEM [61] minimizes background reconstruction error to explicitly decouple style and content, leading to higher editing fidelity. Recent method TextCtrl [56] pre-train a style encoder to enforce style consistency, while GlyphMastero[47] fuses local character-level features with global text-line structures for editing complex Chinese characters. Overall, these methods focus on preserving background information and modifying text content. Although some of them consider text style, they cannot generate text that follows arbitrary external style references. StyleTextGen addresses this limitation

by enabling scene text generation conditioned on any given style reference image.

### 2.3. Image style transfer

Image style transfer aims to apply the style of a reference image to a content image. Early neural style transfer methods [10] formulate this problem as matching deep feature statistics, and subsequent feed-forward approaches [14, 45] enable efficient fast stylization. AdaIN [12] supports arbitrary style transfer by aligning feature statistics, while style-based generative models such as StyleGAN [16, 17] further improve style controllability. More recent work leverages vision–language models and diffusion models for style control: CLIPstyler [19] builds on CLIP [34] to achieve text-guided stylization, diffusion-based methods enable style transfer guided by text or reference images [1, 21, 57], and IP-Adapter [54] injects image-conditional style control into diffusion backbones. These methods mainly target holistic image stylization and are not designed for fine-grained rendering of local text regions with complex glyph structures. In contrast, StyleTextGen focuses on scene text generation conditioned on arbitrary text-style reference images, and provides explicit control over the style of the rendered text.

## 3. Method

### 3.1. Preliminaries

Recent advances in diffusion-based inpainting frameworks, such as FLUX.1-Fill-dev [20] and BrushNet [15], demonstrate strong potential for context-aware image synthesis. Building upon this line of work, we formulate our style-conditioned multilingual scene text generation model under an inpainting paradigm, similar to TextFlux, enabling controllable and high-fidelity transfer of visual text styles from reference images. First, we extract the target text from the given text prompt  $p$ , render it into a binary image  $I_{\text{glyph}}$ , and concatenate vertically it with the scene image  $I_{\text{scene}}$  to provide explicit structural and style cues for text rendering:

$$I_{\text{concat}} = \begin{bmatrix} I_{\text{glyph}} \\ I_{\text{scene}} \end{bmatrix}, \quad (1)$$

where  $I_{\text{glyph}} \in \mathbb{R}^{H \times W \times 1}$  encodes the textual structure and  $I_{\text{scene}} \in \mathbb{R}^{H \times W \times 3}$  provides the background context.

Next, with the inpainting mask  $M$  that indicates the text generation position and a style reference image  $I_{\text{style}}$  that provides style information, we define the conditioning set as  $c = \{I_{\text{concat}}, M, p, I_{\text{style}}\}$ . Therefore, the overall architecture can be formally expressed as:  $\hat{I}_{\text{target}} = \mathcal{D}(\mathcal{F}(\epsilon, c))$ , herein,  $\epsilon$  represents Gaussian noise,  $\mathcal{D}$  denotes the VAE decoder, and  $\mathcal{F}$  stands for the denoising process of DiT. Following the rectified flow [46] view, let  $x_0 = \mathcal{E}(I_{\text{target}})$  denote the clean latent of the target image, and  $\epsilon \sim \mathcal{N}(0, I)$

denote a standard Gaussian noise sample. The forward process adds the noise to  $x_0$  to generate the noisy sample  $x_t$  with the linear noise adding schedule  $\sigma_t$  at time-step  $t$ :

$$x_t = (1 - \sigma_t)x_0 + \sigma_t\epsilon. \quad (2)$$

The DiT backbone predicts a conditional velocity field  $v_\theta(x_t, t, c)$  and is optimized using the conditional flow-matching objective:

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t, x_0, \epsilon} \left[ \omega_t \|v_\theta(x_t, t, c) - (\epsilon - x_0)\|_2^2 \right], \quad (3)$$

where  $\omega_t$  is a time-dependent weighting term.

### 3.2. Overall Architecture

The overall architecture of our proposed StyleTextGen is shown in Figure 2. Given a background scene image, a text prompt and a style reference text image, StyleTextGen could render text within a designated region such that the generated text faithfully inherits the font, color, and texture characteristics of the reference style. In Section 3.3, we present a dual-branch style encoder to tackle the challenge of extracting text styles from complex multilingual scenes, and further explain the injection of text-aware style embeddings for controllable style transfer. To ensure style consistency between the generated text and the reference, we further introduce a text style consistency loss, whose formulation is described in Section 3.4. Furthermore, we propose an inference-time style injection strategy described in Section 3.5, which refines the style consistency of the generated text by adjusting attention responses.

### 3.3. Dual-Branch Style Encoder

To robustly encode fine-grained text style cues under complex multilingual scenes, we design a **Dual-Branch Style Encoder**. Given a style reference image  $I_{\text{style}}$ , the encoder consists of two complementary branches—a *textual style branch* and a *visual prior branch*. Their fused outputs yield a text-aware style embedding  $z_{\text{style}}$  that conditions the DiT backbone, with the detailed architecture illustrated in Figure 2(b).

**Textual style branch.** This branch captures text-specific style cues such as glyph structure, stroke texture, and color distribution. It contains three sequential components: a text-oriented encoder  $E_{\text{text}}$ , a self-attention refinement transformer  $S_{\text{text}}$ , and a Q-Former [23] adaptor  $Q_{\text{text}}$ . The text-oriented encoder is pretrained on a style-preserving multilingual text segmentation task, as in Figure 2(c), encouraging the extraction of stable textual style features from complex backgrounds and diverse scripts. The encoding process is formulated as

$$h_{\text{text}} = Q_{\text{text}}(S_{\text{text}}(E_{\text{text}}(I_{\text{style}}))), \quad (4)$$

where  $h_{\text{text}}$  denotes the refined textual style representation.

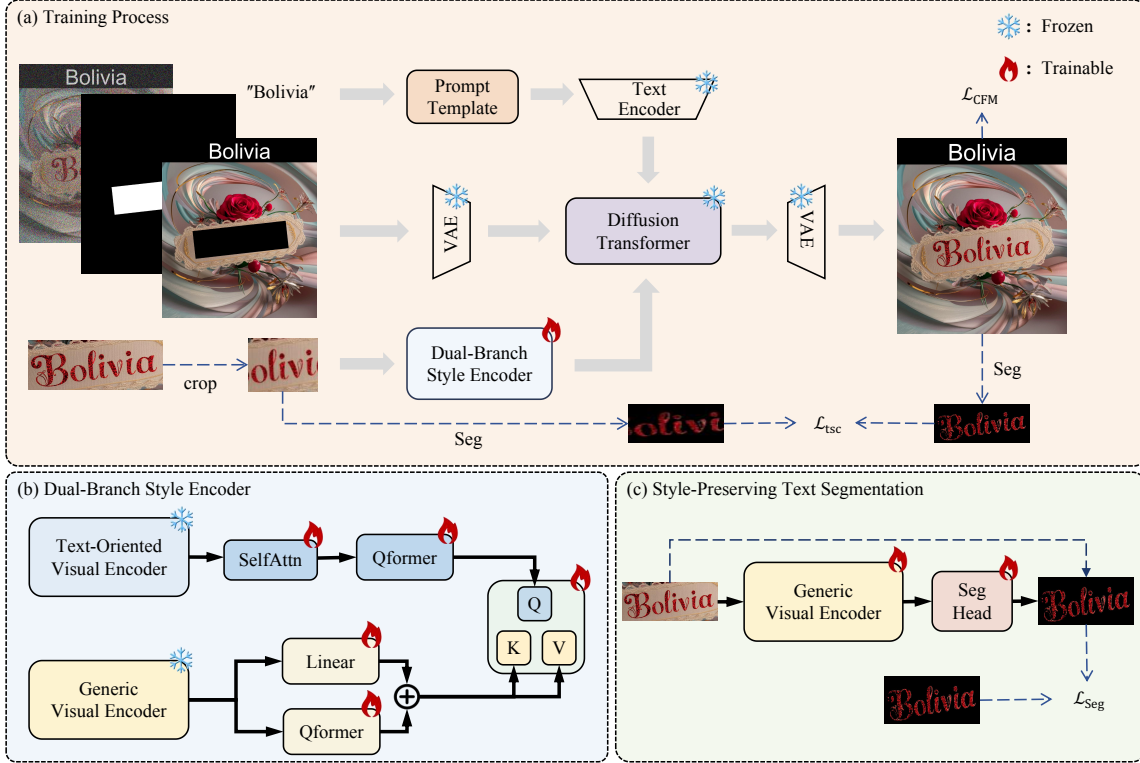


Figure 2. Overview of StyleTextGen. (a) Training process. The inpainting input to the diffusion transformer is constructed from a scene text image concatenated with its glyph map. A randomly cropped text patch serves as the style reference. The model is optimized with the flow-matching loss  $\mathcal{L}_{CFM}$  on the inpainting task and the text style consistency loss  $\mathcal{L}_{tsc}$  computed between the segmented reference and generated text regions. (b) Dual-Branch Style Encoder. The style image is encoded by a text-oriented visual encoder and a generic visual encoder, whose outputs are combined to obtain a text-aware style embedding. (c) Style-Preserving Text Segmentation. This pretraining task targets the text-oriented encoder and preserves text style and layout instead of producing binary masks, encouraging the encoder to learn rich style-aware text features.

**Visual prior branch.** Built upon a generic vision encoder instantiated with a pretrained vision-language backbone [42], this branch extracts holistic visual information to stabilize and generalize the learned style embedding. It employs a visual encoder  $E_{vis}$ , an MLP projector  $P_{vis}$ , and a Q-Former module  $Q_{vis}$ :

$$h_{vis} = P_{vis}(E_{vis}(I_{style})) + Q_{vis}(E_{vis}(I_{style})). \quad (5)$$

This visual prior provides global appearance and tonal coherence complementary to the text-specific features.

The two representations are fused through cross-attention, using  $h_{text}$  as queries and  $h_{vis}$  as keys and values:

$$z_{style} = \text{Attn}(h_{text}, h_{vis}, h_{vis}), \quad (6)$$

producing a robust text-aware style embedding.

Finally, The fused embedding  $z_{style}$  is linearly projected to style key-value pairs  $(K_s, V_s)$ , which modulate the DiT attention through an additive style-attention branch:

$$F_{style} = \text{SelfAttn}(Q, K, V) + \text{StyleAttn}(Q, K_s, V_s), \quad (7)$$

where  $F_{style}$  integrates contextual dependencies from self-attention with style-specific modulation.

### 3.4. Text Style Consistency Loss

Background clutter may obscure style cues, and the absence of an explicit text-region objective often leads to character style drift. We therefore introduce a **Text Style Consistency Loss** that aligns the style statistics of generated text with those of the reference within text regions.

Following the neural style representation approach of Gatys et al. [10], we extract multi-layer features from a pre-trained visual encoder  $\phi$  to capture the style statistics of an image, and let  $J$  be a set of its layers. For layer  $j$ , denote  $\phi_j(x) \in \mathbb{R}^{C_j \times H_j \times W_j}$  and reshape it to  $F_j(x) \in \mathbb{R}^{C_j \times N_j}$  with  $N_j = H_j W_j$ . The Gram matrix, which measures inter-channel correlations and summarizes texture appearance, is computed as:

$$G_j^\phi(x) = \frac{1}{N_j} F_j(x) F_j(x)^\top. \quad (8)$$

The text style consistency loss is then defined as the Frobe-

nus distance between the Gram matrices of the text regions in the generated image and the reference. Given the generated image  $\hat{x}$  with text mask  $M_{\text{gen}}$  and the reference image  $I_{\text{style}}$  with mask  $M_{\text{ref}}$ , the loss is

$$\mathcal{L}_{\text{tsc}} = \sum_{j \in J} \left\| G_j^\phi(M_{\text{gen}} \odot \hat{x}) - G_j^\phi(M_{\text{ref}} \odot I_{\text{style}}) \right\|_F^2. \quad (9)$$

Finally, the overall training objective combines the conditional flow-matching loss and the text style consistency loss as:

$$\mathcal{L} = \mathcal{L}_{\text{CFM}} + \lambda_{\text{tsc}} \mathcal{L}_{\text{tsc}}, \quad (10)$$

where  $\lambda_{\text{tsc}}$  balances the influence of the style consistency term and is set to 10 by default.

### 3.5. Inference-time Style Injection

In this section, we introduce an inference-time style injection strategy that directly refines the attention responses using the reference features.

Given a generated image  $\hat{I}_{\text{gen}}$  and a style reference image  $I_{\text{style}}$ , we first employ a bilingual text segmentation model to extract the text-region masks for both images, denoted as  $M_{\text{gen}}$  and  $M_{\text{style}}$ . These masks enable localized feature manipulation within text areas for subsequent style injection.

Next, we invert the latent feature  $x_s$  of reference image  $I_{\text{style}}$  through DiT to recover its internal key–value features  $(K_s, V_s)$ , where the mask  $M_{\text{style}}$  ensures that only text regions contribute to the extracted style representation. This inversion [25, 37] effectively maps the reference image back to the model’s latent attention space, providing style-aware feature pairs for mask-guided modulation.

Specifically, we first obtain the style-adapted key and value features via AdaIN:

$$\tilde{K}, \tilde{V} = \text{AdaIN}(K, V; K_s \odot M_{\text{style}}, V_s \odot M_{\text{style}}), \quad (11)$$

and then combined with the original ones via  $M_{\text{gen}}$ :

$$\begin{aligned} K' &= (1 - M_{\text{gen}}) \odot K + M_{\text{gen}} \odot \tilde{K}, \\ V' &= (1 - M_{\text{gen}}) \odot V + M_{\text{gen}} \odot \tilde{V}. \end{aligned} \quad (12)$$

Next, under the guidance of these masks, we accurately extract style-related features and ensure that such style control is precisely applied to the text regions of interest:

$$\begin{aligned} f_{\text{base}} &= \text{Attention}(Q, K', V'), \\ f_{\text{style}} &= \text{Attention}(Q, K_s, V_s), \end{aligned} \quad (13)$$

and obtain the final output via mask-based blending:

$$\begin{aligned} f_{\text{out}} &= (1 - M_{\text{gen}}) \odot f_{\text{base}} + \\ &M_{\text{gen}} \odot \text{AdaIN}(f_{\text{base}}; f_{\text{style}} \odot M_{\text{style}}). \end{aligned} \quad (14)$$

To preserve scene-text fidelity and maintain background coherence, we apply the style injection only during the first 10 denoising steps of the generation process.



Figure 3. Qualitative comparison on the StyleText-CE benchmark under the self-style reference setting.

## 4. Experiments

### 4.1. Dataset

**Data synthesis.** Inspired by the ArtText Diffusion model from POSTA [3], which uses an inpainting-based approach to generate stylized text through character-level masks and style prompts, we create a bilingual stylized dataset for model training. We first collect 77 bilingual TrueType font files that support both Chinese and English characters to generate corresponding text masks. Next, we prepare 2,218 artistic style prompts and 10,000 text-free background images. These resources serve as input to the ArtText Diffusion model to synthesize the bilingual style dataset.

**Training data.** For the training data, we begin with 10,000 high-quality bilingual images accompanied by text box pairs as a base dataset. We then follow the data construction pipeline introduced in Calligrapher [30] to generate 2,000 stylized English samples, utilizing the Flux model along with explicit style text prompts. Finally, we select and incorporate an additional 2,000 high-quality synthesized bilingual style images into the final training set.

### 4.2. Benchmark

**AnyWord-Eval.** AnyWord-Eval [43] consists of 1,000 English samples and 1,000 Chinese samples, provides a balanced bilingual testbed.

**StyleText-CE.** To further evaluate the model’s style control capability in cross-lingual scenarios, we construct

Table 1. Comparison with state-of-the-art multi-lingual (English and Chinese) methods.  $\uparrow/\downarrow$  indicates higher/lower is better. Our approach outperforms prior methods on all metrics.

Model	Text Accuracy				Style Similarity			
	Sen.Acc $\uparrow$		NED $\uparrow$		FID $\downarrow$		LPIPS $\downarrow$	
	English	Chinese	English	Chinese	English	Chinese	English	Chinese
AnyText	0.5613	0.4986	0.7264	0.6691	67.82	71.35	0.212	0.267
Calligrapher	0.6129	0.5153	0.7637	0.6847	54.26	63.28	0.176	0.231
TextFlux	0.6542	0.6185	0.8018	0.7934	59.44	57.12	0.187	0.204
Ours	<b>0.7102</b>	<b>0.6524</b>	<b>0.8575</b>	<b>0.8187</b>	<b>49.51</b>	<b>52.43</b>	<b>0.161</b>	<b>0.187</b>

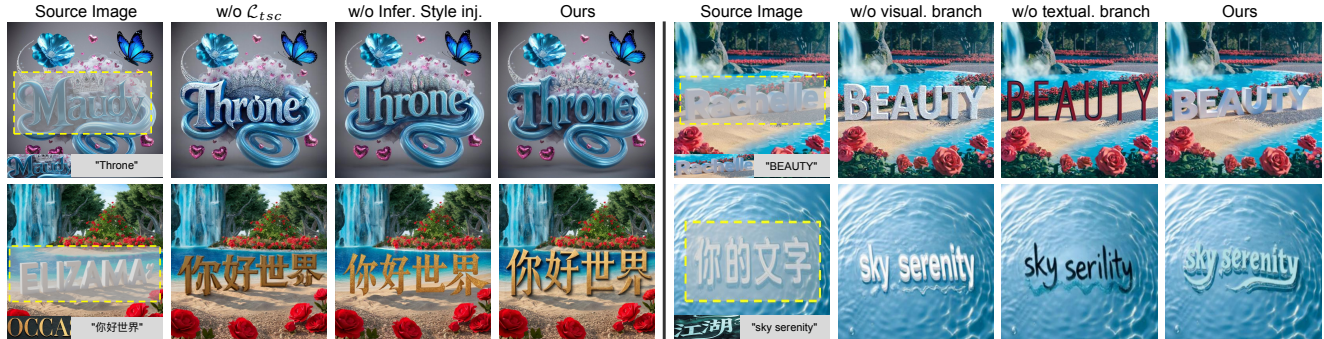


Figure 4. Qualitative results of the ablation study. The left group shows the effects of removing the Text Style Consistency Loss ( $\mathcal{L}_{tsc}$ ) and the inference-time style injection module. The right group presents ablations of the Dual-Branch Style Encoder by removing either the visual prior branch or the textual style branch. The first row corresponds to the self-style reference setting, while the second row shows the external-style reference setting.

StyleText-CE, containing 2021 high-quality Chinese and English scene text images. The benchmark supports evaluation under both self-style and external-style reference modes: in the self-style mode, the style reference image is the original image itself, whereas in the external-style mode, the style pattern is provided by an extra reference image. These two settings allow for comprehensive assessment under both monolingual and cross-lingual configurations.

### 4.3. Implementation Details

**Model Initialization.** For Dual-Branch Style Encoder, the Textual Style Branch adopts InternViT [6] as a text-oriented encoder, initialized with the pretrained weights of TokenFD [11], which has been trained on vision-language alignment tasks involving visual text and demonstrates strong bilingual text perception capabilities. We further perform style-preserving text segmentation training on our synthesized bilingual style dataset to enhance its ability to localize text regions and capture text style attributes. The Visual Prior Branch uses the text-insensitive generic visual encoder SigLIP [42] to provide robust visual representations and avoid shortcuts in the self-reference mode.

Our DIT model is initialized with the weights of TextFlux [49], a multilingual scene text generation model

that demonstrates strong multilingual generation capability and is built upon FLUX.1-Fill-Dev [20].

**Bilingual Text Segmentation Model.** During the computation of the Text Style Consistency Loss and the inference-time style injection, we further fine-tune Hi-SAM[55] on our synthesized bilingual style dataset to enhance its segmentation performance on multilingual text images.

**Training Setup.** During training, we freeze all parameters of the DIT model and the encoder components of both branches in the Dual-Branch Style Encoder. Only the remaining components of the two branches and the linear layers that map the text-aware style embedding to the style key-value pairs are optimized, preserving the representational capacity of the pretrained modules. The model is trained at a resolution of  $512 \times 512$ , using the AdamW [27] optimizer with a learning rate of  $2 \times 10^{-5}$ , a batch size of 2, and 16-step gradient accumulation.

### 4.4. Quantitative Results

For text correctness evaluation, we adopt two widely used metrics: (i) Sen.Acc, which measures line-level text recognition accuracy, and (ii) NED, the normalized edit distance between the generated text and the ground truth. Following [43], we employ the same OCR tool [22] to extract textual content from generated images and compute Sen.Acc. For

Table 2. Quantitative comparison between Calligrapher and StyleTextGen on the StyleText-CE benchmark under different style reference settings. Columns correspond to monolingual (cn, en) and cross-lingual (cn→en, en→cn) configurations.

Model	Self Style Reference				External Style Reference			
	cn	en	cn→en	en→cn	cn	en	cn→en	en→cn
<i>Text Accuracy (Sen.Acc ↑ / NED ↓)</i>								
Calligrapher	0.51 / 0.67	0.64 / 0.78	0.57 / 0.74	0.49 / 0.66	0.49 / 0.67	0.58 / 0.75	0.55 / 0.72	0.47 / 0.64
Ours	<b>0.66 / 0.81</b>	<b>0.70 / 0.85</b>	<b>0.66 / 0.80</b>	<b>0.64 / 0.77</b>	<b>0.60 / 0.79</b>	<b>0.71 / 0.84</b>	<b>0.67 / 0.77</b>	<b>0.65 / 0.82</b>
<i>Style Similarity (FID ↓ / LPIPS ↓)</i>								
Calligrapher	86.24 / 0.44	82.37 / 0.45	134.82 / 0.50	149.68 / 0.52	136.41 / 0.53	105.26 / 0.48	137.94 / 0.55	147.83 / 0.55
Ours	<b>76.95 / 0.42</b>	<b>77.48 / 0.40</b>	<b>124.36 / 0.48</b>	<b>135.27 / 0.51</b>	<b>117.92 / 0.52</b>	<b>93.54 / 0.44</b>	<b>125.88 / 0.54</b>	<b>129.63 / 0.50</b>



Figure 5. Qualitative comparison on the StyleText-CE benchmark under the external-style reference setting.

style similarity assessment, we use: (i) FID, which quanti-

fies the distribution-level discrepancy between visual style features, and (ii) LPIPS, which measures perceptual similarity at the sample level.

We first conduct quantitative analysis on AnyWord-Eval benchmark under the editing (self-style reference) paradigm, as shown in Table 1. Since Calligrapher does not originally support Chinese text generation, we adopt the TextFlux backbone only for the Chinese evaluation, while keeping its original configuration for the English setting. Our method outperforms existing approaches in both text accuracy and style similarity. Specifically, the sentence accuracy of our model exceeds TextFlux by 5.6 and 3.4 percentage points on English and Chinese data, respectively. The normalized edit distance also improves consistently, increasing from 0.8018 to 0.8575 on English and from 0.7934 to 0.8187 on Chinese. In terms of style similarity, our method shows strong style adherence, with FID scores reduced by 16.7% on English data and 8.21% on Chinese data compared with TextFlux. Moreover, our model achieves the lowest LPIPS scores among all methods, decreasing from 0.187 to 0.161 on English and from 0.204 to 0.187 on Chinese, indicating that the generated results are perceptually closer to the reference styles.

To comprehensively evaluate the ability of style-conditioned scene text generation, we further conduct experiments on the StyleText-CE benchmark. Since Calligrapher is the only existing method that supports the external style reference paradigm, we compare our model with it under this setting.

For evaluation on StyleText-CE, the conventional computation of FID is not applicable in the external style reference scenario. Therefore, we employ a bilingual text segmentation model to extract text regions from both the reference and generated images, and compute FID and LPIPS on the segmented regions to avoid background interference.

As shown in Table 2, our method consistently outperforms Calligrapher across all four metrics. Under the self-style reference setting, our model improves Sen.Acc and NED by 0.11 and 0.095, while reducing FID and LPIPS by approximately 9.8 and 0.025, respectively. Under the

Table 3. Quantitative ablation results. We analyze the contributions of the Dual-Branch Style Encoder, the Text Style Consistency Loss ( $\mathcal{L}_{tsc}$ ), and the inference-time style injection module. “text. branch” and “visual. branch” denote the textual style branch and the visual prior branch of the encoder, respectively.

Variant	Sen.Acc $\uparrow$	NED $\uparrow$	FID $\downarrow$	LPIPS $\downarrow$
w/o text. branch	0.618	0.775	133.62	0.536
w/o visual. branch	0.634	0.789	124.18	0.503
w/o $\mathcal{L}_{tsc}$	0.629	0.788	126.94	0.509
w/o infer. style inj.	0.646	0.796	118.36	0.494
<b>Full (Ours)</b>	<b>0.659</b>	<b>0.804</b>	<b>113.47</b>	<b>0.482</b>

external-style reference setting, the gains remain clear: Sen.Acc and NED increase by 0.135 and 0.11, while FID and LPIPS decrease by about 15.1 and 0.028.

It is worth noting that FID and LPIPS values are generally higher in the cross-lingual configurations of the self-style reference setting and in the external style reference setting. This is because style similarity is evaluated only on text regions, where the glyph shapes of the reference text and the generated text often differ significantly, leading to larger perceptual and distribution-level discrepancies.

#### 4.5. Qualitative Results

In the qualitative analysis, we first compare our method with TextFlux[49] and Calligrapher[30] under the self-style reference paradigm (noting that TextFlux does not support reference-based generation—its output style is solely influenced by the background context, and is included here only for visual comparison). As shown in Figure 3, the first two rows present monolingual cases, where our method produces outputs with minimal color deviation from the style reference and achieves noticeably higher style consistency. The last two rows illustrate cross-lingual scenarios, in which our model demonstrates robust style feature extraction and transfer across languages, generating text with higher fidelity and more consistent stylistic alignment. For the external style reference paradigm, Figure 5 presents a comparison between our approach and Calligrapher. From the cases in the first and fifth rows, we observe that our Dual-Branch Style Encoder effectively focuses on text regions, enabling more accurate extraction of style information even under visually cluttered backgrounds. The other examples further show that our method produces more consistent styles, generating text that matches the reference style more closely.

#### 4.6. Ablation Study

In this section, we analyze the contributions of the key components in StyleTextGen. Table 3 reports quantitative results on the StyleText-CE benchmark. We first analyze the

Dual-Branch Style Encoder. Removing either branch results in clear degradation in both text accuracy and style similarity. In particular, removing the textual style branch increases the FID from 113.47 to 133.62, highlighting the importance of accurately extracting text-specific style cues. Removing the visual prior branch also degrades performance, raising the FID to 124.18, which indicates that global visual context provides complementary information for style modeling. Similar trends are observed for Sen.Acc, NED, and LPIPS. We further evaluate the Text Style Consistency Loss and the inference-time style injection strategy. Removing  $\mathcal{L}_{tsc}$  increases the FID to 126.94, demonstrating its role in enforcing alignment between the generated text and the reference style. Disabling the inference-time style injection also degrades performance, increasing the FID to 118.36. These results indicate that both components contribute to improved text fidelity and visual style consistency.

Figure 4 provides qualitative comparisons for the ablation study. The left part shows the effects of removing the Text Style Consistency Loss and the inference-time style injection module. Without the text style consistency loss, the generated text tends to inherit visual cues from surrounding backgrounds rather than from the reference text, leading to inconsistent style transfer. Introducing the loss improves style alignment, while the inference-time style injection further refines visual coherence and produces more stable results. The right part illustrates the ablation of the Dual-Branch Style Encoder. Removing the textual style branch significantly weakens style similarity, particularly under complex backgrounds where accurate extraction of text-specific style cues is required. In contrast, removing the visual prior branch mainly affects glyph shape and layout fidelity. These observations highlight the complementary roles of the two branches in capturing both local textual style and global visual context.

### 5. Conclusion

In this work, we introduced StyleTextGen, a framework for style-conditioned multilingual scene text generation. StyleTextGen employs a dual-branch style encoder that combines a textual style branch with a visual prior branch to learn robust, text-aware style representations across different scripts. To promote consistent text styling, we introduce a text style consistency loss that enforces uniform style within text regions, and an inference-time style injection strategy that further refines style alignment between reference and generated text. We also construct StyleText-CE, a benchmark for evaluating style-conditioned scene text generation. Experiments on AnyWord-Eval and StyleText-CE demonstrate that StyleTextGen improves both text accuracy and style similarity over existing methods, and exhibits strong cross-lingual generalization.

## Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant NO 62376266 and 62406318).

## References

- [1] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023. 3
- [2] Tianjiao Cao, Jiahao Lyu, Weichao Zeng, Weimin Mu, and Yu Zhou. The devil is in fine-tuning and long-tailed problems: a new benchmark for scene text detection. *arXiv preprint arXiv:2505.15649*, 2025. 1
- [3] Haoyu Chen, Xiaojie Xu, Wenbo Li, Jingjing Ren, Tian Ye, Songhua Liu, Ying-Cong Chen, Lei Zhu, and Xinchao Wang. Posta: A go-to framework for customized artistic poster generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28694–28704, 2025. 5
- [4] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser-2: Unleashing the power of language models for text rendering. *arXiv preprint arXiv:2311.16465*, 2023. 2
- [5] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser: Diffusion models as text painters. *arXiv preprint*, abs/2305.10855, 2023. 2
- [6] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024. 6
- [7] Yongkun Du, Zhineng Chen, Caiyan Jia, Xiaoting Yin, Chenxia Li, Yuning Du, and Yu-Gang Jiang. Context perception parallel decoder for scene text recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(6):4668–4683, 2025. 1
- [8] Yongkun Du, Zhineng Chen, Yuchen Su, Caiyan Jia, and Yu-Gang Jiang. Instruction-guided scene text recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(4):2723–2738, 2025. 1
- [9] Zhengyao Fang, Pengyuan Lyu, Jingjing Wu, Chengquan Zhang, Jun Yu, Guangming Lu, and Wenjie Pei. Recognition-synergistic scene text editing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13104–13113, 2025. 2
- [10] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 3, 4
- [11] Tongkun Guan, Zining Wang, Pei Fu, Zhengtao Guo, Wei Shen, Kai Zhou, Tiezhu Yue, Chen Duan, Hao Sun, Qianyi Jiang, et al. A token-level text image foundation model for document understanding. *arXiv preprint arXiv:2503.02304*, 2025. 6
- [12] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017. 3
- [13] Jiabao Ji, Guanhua Zhang, Zhaowen Wang, Bairu Hou, Zhifei Zhang, Brian Price, and Shiyu Chang. Improving diffusion models for scene text editing with dual encoders, 2023. 2
- [14] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 3
- [15] Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In *European Conference on Computer Vision*, pages 150–168. Springer, 2024. 3
- [16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 3
- [17] 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 8110–8119, 2020. 2, 3
- [18] Praveen Krishnan, Rama Kovvuri, Guan Pang, Boris Vasilev, and Tal Hassner. Textstylebrush: transfer of text aesthetics from a single example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):9122–9134, 2023. 2
- [19] Gihyun Kwon and Jong Chul Ye. Clipstyler: Image style transfer with a single text condition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18062–18071, 2022. 3
- [20] Black Forest Labs. Flux. <https://github.com/black-forest-labs/flux>, 2024. 1, 3, 6
- [21] Mingkun Lei, Xue Song, Beier Zhu, Hao Wang, and Chi Zhang. Stylestudio: Text-driven style transfer with selective control of style elements. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23443–23452, 2025. 3
- [22] Chenxia Li, Weiwei Liu, Ruoyu Guo, Xiaoting Yin, Kaitao Jiang, Yongkun Du, Yuning Du, Lingfeng Zhu, Baohua Lai, Xiaoguang Hu, Dianhai Yu, and Yanjun Ma. Pp-ocrv3: More attempts for the improvement of ultra lightweight OCR system. *CoRR*, abs/2206.03001, 2022. 6
- [23] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint*, abs/2301.12597, 2023. 3
- [24] Zhenhang Li, Yan Shu, Weichao Zeng, Dongbao Yang, and Yu Zhou. First creating backgrounds then rendering texts: A new paradigm for visual text blending. *arXiv preprint arXiv:2410.10168*, 2024. 2
- [25] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. 5

- [26] Zeyu Liu, Weicong Liang, Zhanhao Liang, Chong Luo, Ji Li, Gao Huang, and Yuhui Yuan. Glyph-byt5: A customized text encoder for accurate visual text rendering. *arXiv preprint arXiv:2403.09622*, 2024. 2
- [27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 6
- [28] Jiahao Lyu, Wei Wang, Dongbao Yang, Jinwen Zhong, and Yu Zhou. Arbitrary reading order scene text spotter with local semantics guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5919–5927, 2025. 1
- [29] Jian Ma, Yonglin Deng, Chen Chen, Nanyang Du, Haonan Lu, and Zhenyu Yang. Glyphdraw2: Automatic generation of complex glyph posters with diffusion models and large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5955–5963, 2025. 2
- [30] Yue Ma, Qingyan Bai, Hao Ouyang, Ka Leong Cheng, Qiyu Wang, Hongyu Liu, Zichen Liu, Haofan Wang, Jingye Chen, Yujun Shen, et al. Calligrapher: Freestyle text image customization. *arXiv preprint arXiv:2506.24123*, 2025. 2, 5, 8
- [31] OpenAI. Dall-e3. <https://openai.com/index/dall-e-3/>, 2023. 2
- [32] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024. 1
- [33] 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. 2
- [34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 3
- [35] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, pages 8821–8831. PMLR, 2021. 2
- [36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 2
- [37] Litu Rout, Yujia Chen, Nataniel Ruiz, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Semantic image inversion and editing using rectified stochastic differential equations. *arXiv preprint arXiv:2410.10792*, 2024. 5
- [38] Prasun Roy, Saumik Bhattacharya, Subhankar Ghosh, and Umapada Pal. Stefann: scene text editor using font adaptive neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13228–13237, 2020. 2
- [39] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022. 2
- [40] Yan Shu, Hangui Lin, Yexin Liu, Yan Zhang, Gangyan Zeng, Yan Li, Yu Zhou, Ser-Nam Lim, Harry Yang, and Nicu Sebe. When semantics mislead vision: Mitigating large multimodal models hallucinations in scene text spotting and understanding. *arXiv preprint arXiv:2506.05551*, 2025.
- [41] Yan Shu, Weichao Zeng, Fangmin Zhao, Zeyu Chen, Zhenhang Li, Xiaomeng Yang, Yu Zhou, Paolo Rota, Xiang Bai, Lianwen Jin, et al. Visual text processing: A comprehensive review and unified evaluation. *arXiv preprint arXiv:2504.21682*, 2025. 2
- [42] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 4, 6
- [43] Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Geng, and Xuansong Xie. Anytext: Multilingual visual text generation and editing. *arXiv*, 2023. 2, 5, 6
- [44] Yuxiang Tuo, Yifeng Geng, and Liefeng Bo. Anytext2: Visual text generation and editing with customizable attributes, 2024. 2
- [45] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. *arXiv preprint arXiv:1603.03417*, 2016. 3
- [46] Fu-Yun Wang, Ling Yang, Zhaoyang Huang, Mengdi Wang, and Hongsheng Li. Rectified diffusion: Straightness is not your need in rectified flow, 2024. 3
- [47] Tong Wang, Ting Liu, Xiaochao Qu, Chengjing Wu, Luoqi Liu, and Xiaolin Hu. Glyphmastero: A glyph encoder for high-fidelity scene text editing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28523–28532, 2025. 2
- [48] Liang Wu, Chengquan Zhang, Jiaming Liu, Junyu Han, Jingtuo Liu, Errui Ding, and Xiang Bai. Editing text in the wild. In *Proceedings of the 27th ACM international conference on multimedia*, pages 1500–1508, 2019. 2
- [49] Yu Xie, Jielei Zhang, Pengyu Chen, Ziyue Wang, Weihang Wang, Longwen Gao, Peiyi Li, Huyang Sun, Qiang Zhang, Qian Qiao, et al. Textflux: An ocr-free dit model for high-fidelity multilingual scene text synthesis. *arXiv preprint arXiv:2505.17778*, 2025. 6, 8
- [50] Qiangpeng Yang, 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 14700–14709, 2020. 2
- [51] Xiaomeng Yang, Zhi Qiao, and Yu Zhou. Ipad: iterative, parallel, and diffusion-based network for scene text recognition. *International Journal of Computer Vision*, 133(8): 5589–5609, 2025. 1
- [52] Yukang Yang, Dongnan Gui, Yuhui Yuan, Haisong Ding, Han Hu, and Kai Chen. Glyphcontrol: Glyph condi-

- tional control for visual text generation. *arXiv preprint*, abs/2305.18259, 2023. 2
- [53] Zhoufaran Yang, Yan Shu, Jing Wang, Zhifei Yang, Yan Zhang, Yu Li, Keyang Lu, Gangyan Zeng, Shaohui Liu, Yu Zhou, et al. Vidtext: Towards comprehensive evaluation for video text understanding. *arXiv preprint arXiv:2505.22810*, 2025. 2
- [54] Hu Ye, Jun Zhang, Sibao Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint*, 2023. 3
- [55] Maoyuan Ye, Jing Zhang, Juhua Liu, Chenyu Liu, Baocai Yin, Cong Liu, Bo Du, and Dacheng Tao. Hi-sam: Marrying segment anything model for hierarchical text segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–16, 2024. 6
- [56] Weichao Zeng, Yan Shu, Zhenhang Li, Dongbao Yang, and Yu Zhou. Textctrl: Diffusion-based scene text editing with prior guidance control. *Advances in Neural Information Processing Systems*, 37:138569–138594, 2024. 2
- [57] Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Inversion-based style transfer with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10146–10156, 2023. 3
- [58] Qilong Zhangli, Jindong Jiang, Di Liu, Licheng Yu, Xiaoliang Dai, Ankit Ramchandani, Guan Pang, Dimitris N. Metaxas, and Praveen Krishnan. Layout agnostic scene text image synthesis with diffusion models, 2024. 2
- [59] Yiming Zhao and Zhouhui Lian. Udifftext: A unified framework for high-quality text synthesis in arbitrary images via character-aware diffusion models, 2023. 2
- [60] Tianlun Zheng, Zhineng Chen, Shancheng Fang, Hongtao Xie, and Yu-Gang Jiang. Cdistnet: Perceiving multi-domain character distance for robust text recognition. *IJCV*, 132(2): 300–318, 2024. 1
- [61] Jianqun Zhou, Pengwen Dai, Yang Li, Manjiang Hu, and Xiaochun Cao. Explicitly-decoupled text transfer with the minimized background reconstruction for scene text editing. *IEEE Transactions on Image Processing*, 2024. 2