Towards Training-Free Scene Text Editing

Yubo Li[2,3,4*], Xugong Qin[1,*], Peng Zhang[1,†], Hailun Lin[2,3,], Gangyan Zeng[1], Kexin Zhang[1]
[1]School of Cyber Science and Engineering, Nanjing University of Science and Technology
[2]Institute of Information Engineering, Chinese Academy of Sciences
[3]State Key Laboratory of Cyberspace Security Defense
[4]School of Cyber Security, University of Chinese Academy of Sciences
liyubo2023@iie.ac.cn, qinxugong@njust.edu.cn

Abstract

Scene text editing seeks to modify textual content in natural images while maintaining visual realism and semantic consistency. Existing methods often require task-specific training or paired data, limiting their scalability and adaptability. In this paper, we propose TextFlow, a training-free scene text editing framework that integrates the strengths of Attention Boost (AttnBoost) and Flow Manifold Steering (FMS) to enable flexible, high-fidelity text manipulation without additional training. Specifically, FMS preserves the structural and style consistency by modeling the visual flow of characters and background regions, while AttnBoost enhances the rendering of textual content through attention-based guidance. By jointly leveraging these complementary modules, our approach performs end-to-end text editing through semantic alignment and spatial refinement in a plug-and-play manner. Extensive experiments demonstrate that our framework achieves visual quality and text accuracy comparable to or superior to those of training-based counterparts, generalizing well across diverse scenes and languages. This study advances scene text editing toward a more efficient, generalizable, and training-free paradigm. Code is available at https://github.com/lyb18758/TextFlow

1. Introduction

Scene Text Editing (STE) [35, 36, 48] aims to modify or replace text in natural images while preserving background and key visual attributes of the original text, including font style, color, size, and geometric layout. This task has broad practical value in applications such as image translation [44], advertisement design [57], content-aware image editing [52], data augmentation for text recognition [9, 28], and other text-centric vision tasks [11–13, 29–34, 39, 53].

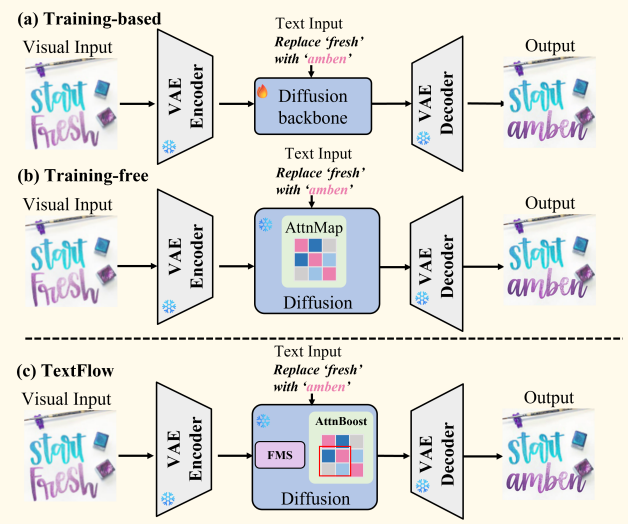

Figure 1. Comparison of the pipelines between training-based and training-free methods for scene text editing. Training-based methods require large-scale, high-quality paired data that require high computing resources. The training-free method mostly focuses on the attention map for general objects, but ignores the text accuracy and style consistency.

Generative models have evolved significantly, from early Generative Adversarial Networks (GANs) [7, 19, 37, 51, 59] that faced training instability, to UNet-based diffusion models [4, 15, 17, 40–42] that improved output fidelity and diversity, and further to Diffusion Transformers (DiT) [10, 21–23, 50] that enhanced global semantic modeling through Multimodal Attention. These advances have propelled progress in STE, with methods such as DiffSTE [17], AnyText [40], and textFlux [50] demonstrating strong text-rendering performance.

However, a fundamental trade-off exists between adaptability and editing quality. Training-based methods, like Fig. 1(a), require large-scale, high-quality paired data, which is scarce in practice. While synthetic data can supplement training, it often limits generalization to diverse real scenes. Additionally, these approaches demand substan-

*Equal contribution. †Corresponding author.

tial computational resources, restricting their practical use. Training-free methods, as shown in Fig. 1(b), leverage pre-trained models without fine-tuning, with many approaches utilizing attention manipulation for editing tasks. While effective for general object editing, these methods face particular challenges in scene text editing. Preserving precise typographic and structural details in complex scenes with diverse backgrounds, fonts, or layouts remains challenging for attention-based methods, often resulting in visual artifacts and character distortions.

A key limitation of training-free methods lies in their phase-dependent controllability, which arises from the non-uniform signal-to-noise ratio across diffusion timesteps. During early denoising, existing techniques fail to preserve the structural and stylistic foundations, resulting in unstable editing trajectories. In later stages, inadequate semantic and spatial guidance leads to textual inaccuracies, such as character duplication, missing elements, or distortion, thereby hindering coherent text generation.

To address these challenges, we propose TextFlow, a training-free framework for scene text editing. As illustrated in Fig. 1(c), TextFlow introduces phase-aware guidance that separately optimizes style preservation and textual accuracy. Specifically, it operates in two phases: the first employs a Flow Manifold Steering (FMS) module to maintain style consistency, while the second leverages an Attention Boost (AttnBoost) mechanism to improve textual accuracy. Despite requiring no training, our method narrows the performance gap with training-based approaches, achieving competitive editing quality through a single forward pass without task-specific fine-tuning, paired datasets, or resource-intensive retraining. This makes TextFlow both efficient and practical for real-world applications. The main contributions of this work can be summarized as follows:

- We introduce **Flow Manifold Steering (FMS)** module, which operates source and target conditions in the latent space, guiding the denoising trajectory to maintain structural and stylistic consistency from the denoising steps.
- We propose an **Attention Boost (AttnBoost)** mechanism that leverages attention maps to enhance fine-grained text rendering. By dynamically amplifying text-relevant regions during sampling, AttnBoost significantly improves textual accuracy and semantic alignment.
- Through extensive experiments on benchmark datasets, we demonstrate that TextFlow achieves state-of-the-art performance in both visual quality and textual correctness, without any task-specific fine-tuning.

2. Related Work

2.1. Diffusion-Based Scene Text Editing

The widespread application of the UNet-based diffusion model in image editing has driven the development of STE. DiffSTE [17] employs a dual-encoder design with character and instruction encoding to learn the mapping from textual instructions to corresponding images with specified styles in the background; TextDiffuser [5] systematically decouples layout planning from content generation by employing a dual-stage framework; DiffUTE [4] utilizes character glyphs and text positions from the source image as auxiliary information to provide better control during character generation; UDiffText [56] leverages large-scale training data and text embeddings to improve text-based image editing; AnyText [40] encodes auxiliary information such as text glyphs, positions, and mask images into a latent space to assist in text generation and editing; AnyText2 [41] proposes a WriteNet+AttnX architecture, enabling the model to focus more on font and color attributes; DreamText [46] effectively mitigates issues of character repetition, omission, and distortion encountered by existing methods; TextCtrl [54] decomposes the prerequisites of STE into fine-grained style disentanglement and glyph structure representation, integrating style-structure guidance with diffusion models to enhance rendering accuracy and style fidelity; GlyphMastero [45] targets editing tasks with complex characters, such as Chinese, by combining local character-level features and global text-line structures.

To further enhance generation performance, recent studies integrate large-scale transformer architectures as the backbone of diffusion models, resulting in advanced models like DiT [27]. Stable Diffusion 3 [10] and FLUX [21], both based on the flow matching method, have extended the DiT architecture to MM-DiT to achieve superior generation quality. Their subsequent open-source release has provided a significantly more robust foundation for STE. textFlux [50] eliminates the need for OCR encoders; FLUX-Text [23] enhances glyph understanding and generation through lightweight Visual and Text Embedding Modules; Flux-kontext [22] generates novel output views by incorporating semantic context from text and image inputs; Qwen-image [47] separately feed the original image into Qwen2.5-VL and the VAE encoder to obtain semantic and reconstructive representations; HunYuanImage3.0 [3] unifies multimodal understanding and generation within an autoregressive framework. Moreover, GPT-4o Image [26], Gemini 2.5 Flash Image, and Blip3o-NEXT [6] leverage a hybrid Diffusion-Autoregressive architecture to attain state-of-the-art capabilities in image understanding, generation, and editing.

While obtaining exceptional performance on STE tasks, existing methods typically demand considerable resources to solve the challenging problem of editing.

2.2. Training-Free Image Editing

Benefiting from the rapid advancement of the DiT backbone and flow matching techniques, foundation models

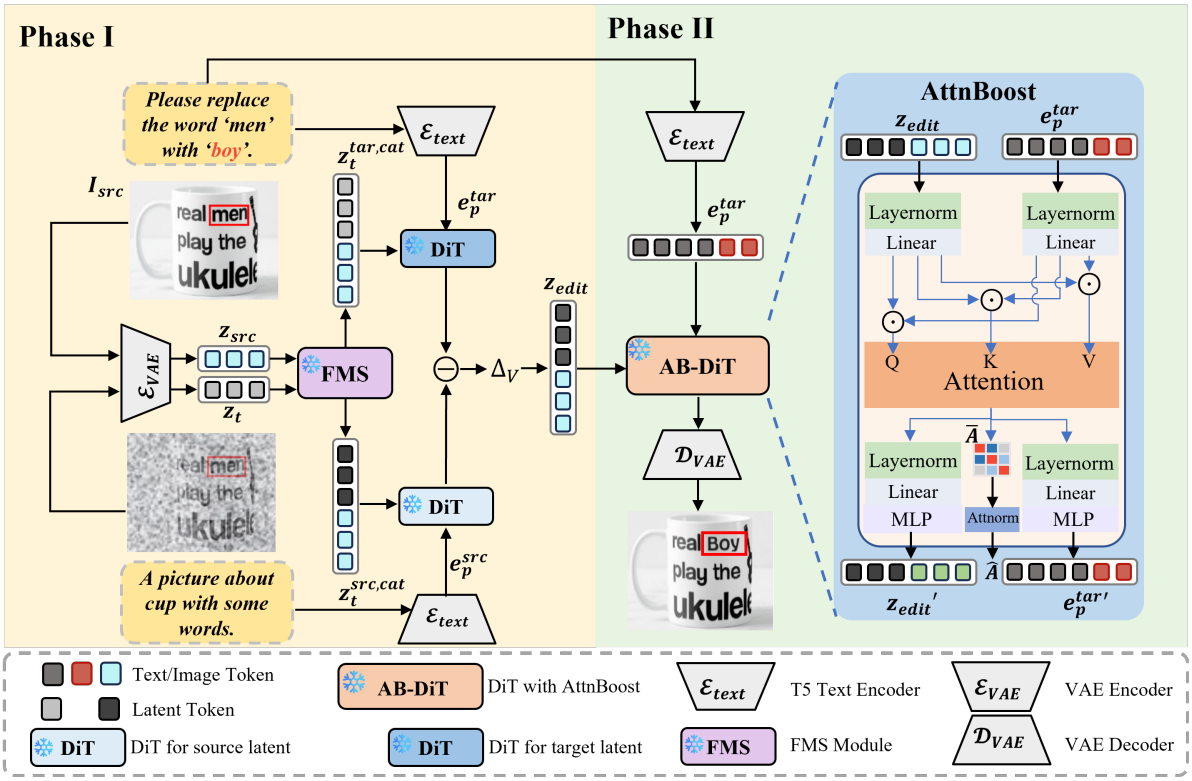

Figure 2. The overall framework of TextFlow. In the first phase, the source image is encoded into latent representations $\mathbf{z}_t$ and $\mathbf{z}_{src}$ via the VAE encoder, which are subsequently processed by the FMS module to generate concatenated representations $\mathbf{z}_t^{src,cat}$ and $\mathbf{z}_t^{tar,cat}$. These representations, along with their corresponding text embeddings $\mathbf{e}_p^{src}$ and $\mathbf{e}_p^{tar}$, are fed into parallel DiT blocks to compute the velocity field differential Δ_V, ultimately producing the edited latent representation $\mathbf{z}_{edit}$; In the second phase, $\mathbf{z}_{edit}$ and the target embedding $\mathbf{e}_p^{tar}$ are processed by the AttnBoost DiT (AB-DiT), where concatenation and self-attention operations generate refined text-to-image attention maps that enhance textual rendering accuracy through spatial-aware amplification.

have demonstrated significantly enhanced generation and editing capabilities alongside robust general-purpose performance. Building upon this progress, there is increasing research interest in exploring training-free methods to further improve the image editing proficiency of these models.

Stable Flow [1] introduce an improved image inversion method for flow models to enable image editing; CannyEdit [49] propose selective canny control and dual-prompt guidance to balance text adherence in edited regions, context fidelity in unedited areas, and seamless integration of edits; ICEdit [55] adopt a diptych framework for both T2I-DiT and inpainting-DiT to achieve in-context editing; KV-Edit [60] uses KV cache in DiTs to maintain background consistency, ultimately generating new content that seamlessly integrates with the background within user-provided regions; RF-Solver [43] proposes a novel training-free sampler that effectively enhances inversion precision by mitigating errors in the ordinary differential equation (ODE) solving process of rectified flow; FlowEdit [20] constructs a direct path between the source and target distributions by breaking away from the editing-by-inversion paradigm; LanPaint [58] propose a training-free, asymptotically exact partial conditional sampling methods for ODE-based and rectified flow models.

Furthermore, building upon these general frameworks, visual text rendering and generation have also seen significant advancements. Specifically, AMO [16] introduce an overshooting sampler for pretrained rectified flow (RF) models, by alternating between over-simulating the learned ODE and reintroducing noise, which improves the text rendering accuracy without compromising image quality; TextCrafter [8] focusing on complex visual text generation, employs a progressive strategy to decompose complex visual text into distinct components while ensuring robust alignment between textual content and its visual carrier.

These methods perform outstandingly in general editing and text rendering. However, for the STE task, there is a distinct lack of research dedicated to training-free methods.

3. Methodology

In this section, we explore training-free editing capabilities within DiT generative models and propose our fusion edit

framework for scene text editing. Our fusion framework is based on the flow matching architecture, a continuous-time generative model that aims to learn a velocity field $v_t(x)$, such that the ODE trajectory defined by this field maps noise $\epsilon \sim \mathcal{N}(0, I)$ to the data sample x. Building upon FLUX-Kontext [22] implemented via flow matching, our approach introduces an innovative two-phased strategy, achieving high-precision scene text editing with low computational cost.

3.1. Overall Framework

The overall pipeline of our proposed TextFlow for denosing steps is illustrated in Fig. 2. Our core insight is to decouple the complex STE task into two complementary phases, each governed by a specialized mechanism to address its unique challenges: **style preservation** and **detail rendering** during the denoising step.

Given a source image I_{src} with its corresponding caption T_{src} and a target text prompt T_{tar}, the process begins by encoding the image into a latent representation to $\mathbf{z}_t$ and $\mathbf{z}_{src}$, processing both texts through a text encoder to obtain their embeddings $\mathbf{e}_p^{src}$ and $\mathbf{e}_p^{tar}$. The denoising trajectory, governed by a pre-trained flow matching model, is then strategically manipulated by our two novel components:

- **FMS module:** Operating in the first phase, as shown in Fig. 2, this module is responsible for establishing and preserving the foundational style and structure of the source image. The outputs compute a velocity field differential $\mathbf{V}_\Delta$ between the source and target trajectories in the latent space and apply a controlled shift, ensuring that the global attributes (e.g., font style, background texture) are coherently retained early in the generation process.
- **AttnBoost mechanism:** Activated in the second phase, as shown in Fig. 2, this mechanism ensures the accurate spelling, legibility, and semantic alignment of the generated text. It extracts and processes the attention maps from the double-stream transformer block, generating a fine-grained guidance signal $\hat{A}$ that directs the scheduler to render text details that precisely match the target description T_{tar}.

3.2. Style Preservation with FMS

During the first phase of the denoising cycle, as shown in Fig. 3, we introduce the FMS module to achieve robust style preservation. This approach operates by manipulating trajectories in the latent space, ensuring structural integrity while accommodating stylistic transformations throughout the editing process.

The core framework of FMS consists of the following three steps. First, we define the parameter controlling noise injection intensity:

$$t_i = \sigma_{\text{step}}[i], \tag{1}$$

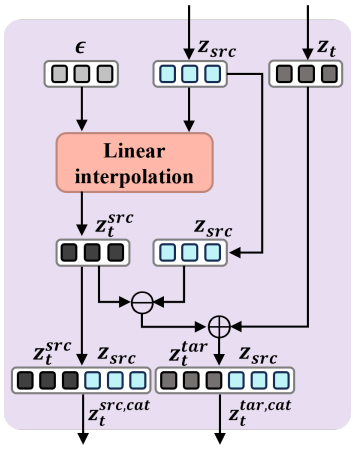

Figure 3. Illustration of the proposed FMS Model. The latent representations $\mathbf{z}_t$ and $\mathbf{z}_{src}$ are processed with random noise ϵ through linear interpolation and vector arithmetic operations to maintain style consistency.

where t_i represents the noise level at the current timestep, and σ_{step} denotes the standard deviation parameter from the diffusion scheduler.

Next, we construct the noise-injected source latent representation:

$$\mathbf{z}_t^{\text{src}} = (1 - t_i) \cdot \mathbf{z}_{\text{src}} + t_i \cdot \epsilon, \tag{2}$$

where $\mathbf{z}_{\text{src}}$ is the original latent representation of the source image, $\mathbf{z}_t^{\text{src}}$ is the noise-injected latent state, and ϵ represents random noise following a standard normal distribution.

We then correct the target latent representation through differential geometric transformation:

$$\mathbf{z}_t^{\text{tar}} = \mathbf{z}_t + (\mathbf{z}_t^{\text{src}} - \mathbf{z}_{\text{src}}), \tag{3}$$

where $\mathbf{z}_t$ is the current latent state of target generation, and $\mathbf{z}_t^{\text{tar}}$ is the corrected target representation. The differential term $(\mathbf{z}_t^{\text{src}} - \mathbf{z}_{\text{src}})$ precisely captures the geometric offset induced by noise injection.

To integrate information, we concatenate the processed states:

$$\mathbf{z}_t^{\text{src,cat}} = \text{Concat}(\mathbf{z}_t^{\text{src}}, \mathbf{z}_t), \tag{4}$$

$$\mathbf{z}_t^{\text{tar,cat}} = \text{Concat}(\mathbf{z}_t^{\text{tar}}, \mathbf{z}_t). \tag{5}$$

Furthermore, we compute the trajectory-shifting vector field for fine-grained control:

$$\mathbf{V}_\Delta = \mathcal{F}\left(\mathbf{z}_t^{\text{src,cat}}, \mathbf{z}_t^{\text{tar,cat}}, \mathbf{e}_p^{\text{src}}, \mathbf{e}_p^{\text{tar}}\right), \tag{6}$$

$$\mathcal{F} := \Phi(z_t^{tar,cat}, e_p^{tar}) - \Phi(z_t^{src,cat}, e_p^{src}), \tag{7}$$

where $\mathcal{F}$ is the velocity field computation function that performs cross-modal feature alignment between source and

target embeddings. Φ represents the standard DiT backbone. Based on this differential, we apply trajectory shifting as follows:

$$\mathbf{z}_{\text{edit}} = \mathbf{z}_t + \mathbf{V}_\Delta \cdot (t_{i-1} - t_i)\,, \tag{8}$$

where t_{i-1} and t_i represent adjacent noise levels in the diffusion process.

This mathematical framework embeds structural preservation constraints into the generation trajectory through rigorous geometric operations, ensuring style coherence while supporting flexible text adaptation, thereby providing a theoretical foundation for training-free scene text editing.

3.3. Detail Rendering by AttnBoost

During the second phase of the denoising cycle, as shown in Fig. 2, we deploy the AttnBoost mechanism to achieve fine-grained text-guided rendering. This module strategically enhances text-relevant regions in the latent space by processing cross-attention maps from the double-stream transformer block. The query (Q), key (K), and value (V) matrices are derived from the concatenation of the edited latent representation $\mathbf{z}_{\text{edit}}$ and the target text embeddings $\mathbf{e}_p^{\text{tar}}$, followed by linear projections through the transformer layers. This ensures precise semantic alignment with target descriptions while maintaining visual consistency with the source image structure.

Our attention computation begins with the standard scaled dot-product formulation:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V, \tag{9}$$

Text Region Enhancement applies targeted amplification to text regions through element-wise transformation:

$$A_{\text{enhanced}}(b, h, q, k) = \begin{cases} \mathcal{T}(A(b, h, q, k)) & \text{if } q \in [\text{start}_1, \text{end}_1], \\ A(b, h, q, k) & \text{otherwise}, \end{cases} \tag{10}$$

where $A \in \mathbb{R}^{B\times H\times L\times S}$ denotes the original attention tensor with batch size B, attention heads H, query length L, and key sequence length S. The transformation function $\mathcal{T} : \mathbb{R} \to \mathbb{R}$ implements the region-specific amplification.

Attention Mapping and Aggregation extracts text-to-image attention patterns and consolidates them through dimensional reduction:

$$A_{\text{t2i}} = A_{\text{enhanced}}[\cdot, \cdot, \mathcal{I}_{\text{text}}, \mathcal{I}_{\text{image}}], \tag{11}$$

$$A_{\text{agg}} = \sum_{q\in\mathcal{I}_{\text{text}}} A_{\text{t2i}}[\cdot, \cdot, q, \cdot], \tag{12}$$

where $\mathcal{I}_{\text{text}} = [\text{start}_1, \text{end}_1]$ represents the text token indices, $\mathcal{I}_{\text{image}} = [N_{\text{text}}, S]$ denotes the image token indices, and N_{text} indicates the quantity of text tokens in the input token sequence.

The extracted attention maps are further refined through spatial pooling, enabling the aggregation of local features and enhancing the focus on relevant regions:

$$\bar{A} = \frac{1}{B \times H \times W} \sum_{i=1}^{B} \sum_{j=1}^{H} \sum_{k=1}^{W} A_{i,j,k}, \tag{13}$$

where $\bar{A}$ represents the spatially pooled attention map, obtained by averaging the original attention tensor A across batch, height, and width dimensions, with W denoting the feature map width.

Normalization is then applied to ensure consistent value ranges and enhance numerical stability:

$$\hat{A} = \frac{\bar{A} - \min(\bar{A})}{\max(\bar{A}) - \min(\bar{A}) + \epsilon}, \quad \epsilon = 1 \times 10^{-8}, \tag{14}$$

where $\hat{A}$ denotes the normalized attention map constrained to $[0, 1]$ range, while ϵ provides numerical stability to prevent division by zero.

The refined attention guidance is integrated into the denoising process through scheduler modulation:

$$z_{t-1} = \mathcal{S}(z_t, \hat{A}, t), \tag{15}$$

where z_t and z_{t-1} represent the latent representations at current and subsequent timesteps, while $\mathcal{S}$ indicates the modified scheduler function that incorporates attention guidance at denoising step t. Further details regarding the $\mathcal{S}$ scheduler and its control enhancement through $\hat{A}$ will be elaborated in the Appendix.

AttnBoost establishes a mathematically grounded framework for transforming cross-modal attention patterns into spatial guidance signals. This systematic processing pipeline, from targeted region enhancement through normalized spatial guidance, enables precise text-controlled rendering while preserving structural integrity, providing a robust foundation for semantically aware image editing in complex visual environments.

4. Experiments

4.1. Datasets and metrics

Datasets. To provide assessments on both image generation quality and visual text quality, we employ the ScenePair dataset [54], a real-world scene text image-pair dataset. Specifically, ScenePair comprises 1,280 image pairs with text labels sourced from ICDAR 2013 [18], HierText [24], and MLT 2017 [25]. Each pair consists of two cropped text images that share similar text length, style, and background, along with the corresponding original full-size images. To ensure consistent input dimensions across all models, we pad the cropped images with background-similar colors to

Table 1. Performance comparison of different methods on the ScenePair dataset.

Methods	ScenePair					
	SSIM ($\times 10^{-2}$) ↑	PSNR ↑	MSE ($\times 10^{-2}$) ↓	FID ↓	ACC (%) ↑	NED ↑
DiffSTE [17]	22.76	12.26	7.34	180.15	71.11	0.907
TextDiffuser [5]	26.99	13.93	5.70	56.67	51.48	0.719
AnyText [40]	30.73	13.66	6.05	51.44	51.12	0.734
TextFlux [50]	86.57	17.96	1.83	54.64	**80.40**	0.911
Flux-fill [21]	82.73	17.10	2.99	107.83	13.74	0.306
Flux-Kontext [22]	87.08	20.53	1.58	15.41	78.72	**0.920**
Qwen-image [47]	77.89	15.14	4.19	56.71	68.59	0.833
FlowEdit [20]	87.60	20.89	1.16	25.41	45.51	0.590
TextFlow (Ours)	**89.03**	**22.47**	**0.91**	**13.53**	79.98	0.914

Figure 4. Qualitative Analysis. The compared methods include both training-based STE approaches like DiffSTE [17], AnyText [40], TextFlux [50] and recent training-free editing techniques FlowEdit [20]. We also include the powerful foundational model Flux-Kontext [22] (F-Kontext), for a more extensive comparison.

a resolution of 384×256, and all metrics are computed based on this preprocessed input.

Evaluation Metrics. For the assessment of image generation quality, we employ the following metrics: (1) Structural Similarity Index Measure (SSIM): Measures the structural similarity between the generated image and the Ground Truth (GT); (2) Peak Signal-to-Noise Ratio (PSNR): calculate the peak signal-to-noise ratio to assess the distortion level by computing the mean squared error between the generated image and the GT; (3) Mean Squared Error (MSE): Quantifies the pixel-wise difference between the generated image and the GT; (4) Fréchet Inception Distance (FID): Evaluates the quality of synthesized images by comparing the statistical distributions of feature embeddings from the generated and GT images. For visual text quality assessment, we utilize Accuracy (ACC) and Normalized Edit Distance (NED) [14] to evaluate the correctness and overall quality of the generated text image, using an official text recognition algorithm [2] and the corresponding checkpoint.

4.2. Implementation Details

Our proposed TextFlow framework is built upon the FLUX-Kontext [22] model as the core image editing generator due to its superior performance in generating high-quality images. For the text encoder, we utilize the T5 and CLIP to extract text embeddings, which provide a robust semantic representation for both the source and target prompts. The entire framework operates in a training-free manner, and no components are fine-tuned on any scene text editing datasets. During the inference process, we employ the Overshoot [16] and Euler scheduler with 50 denoising steps to balance generation quality and computational efficiency. All experiments are performed on a server equipped with 4 NVIDIA A6000 GPUs with 48G VRAM each. Additional experimental settings and implementation details will be provided in the Appendix.

4.3. Comparison with State-of-the-Art Methods

Quantitative Analysis. We conduct a comprehensive evaluation of our proposed TextFlow framework against

state-of-the-art methods on the ScenePair dataset. As summarized in Table 1, the compared methods include both training-based STE approaches like DiffSTE [17], TextDiffuser [5], AnyText [40], TextFlux [50] and recent training-free editing techniques FlowEdit [20]. We also include the powerful foundational model Flux-fill [21], Flux-Kontext [22], and Qwen-image [47] for a more extensive comparison.

The experimental results demonstrate the superior performance of our method across multiple dimensions. In terms of image quality and structural fidelity, our approach achieves the highest SSIM score of 89.03 and the best PSNR of 22.47, significantly outperforming all competing methods. Notably, our method reduces the MSE to 0.91, approximately 42% lower than the second-best method, Flux-Kontext [22], indicating superior pixel-level reconstruction accuracy. The lowest FID score of 13.53 further confirms that our generated images are statistically closest to the real data distribution, highlighting exceptional visual realism.

Regarding textual rendering accuracy, our method achieves a competitive character-level accuracy of 79.98% and NED score of 0.914. While TextFlux [50] shows a slightly higher accuracy of 80.40%, our method maintains a better balance between textual correctness and visual quality, as evidenced by our substantially superior FID and PSNR metrics. This balanced performance is practically crucial for real-world applications where both textual accuracy and visual coherence are paramount. A comprehensive experimental evaluation of additional methods will be provided in the Appendix.

Qualitative Analysis. Fig. 4 presents a qualitative comparison of generated results. Our proposed TextFlow is evaluated against several representative methods, including UNet-based approaches such as DiffSTE [17] and AnyText [40], as well as state-of-the-art DiT-based methods in STE like TextFlux [50], FLUX-Kontext [22], and FlowEdit [20]. For methods requiring mask-conditioned inputs, such as AnyText [40] and TextFlux [50], we applied background-colored padding to the input images to maintain consistent input resolution. Regarding prompt design, the source description was uniformly formatted as: *"A picture with word 'T_{src}'."*, while the target prompt followed the structured template: *"Please replace the word 'T_{src}' with 'T_{tar}'."*.

While TextFlux [50] maintains relatively high text accuracy, it suffers from significant style loss. Conversely, FLUX-Kontext [22] demonstrates better style preservation but shows deficiencies in text accuracy. FlowEdit [20], as a training-free approach, achieves reasonable performance in both style consistency and text accuracy, yet falls short in handling fine-grained details such as letter case consistency and glyph structure. In contrast, as demonstrated in the fifth row with the word "*Servicemenu*" and the sixth row with "*Smooth*", our method achieves superior performance in both style preservation and text accuracy while maintaining excellent detail handling capabilities.

Fig. 5 shows editing results on full-size images, where TextFlow achieves competitive performance in style preservation and text accuracy against other DiT-based methods, underscoring its superior editing capability.

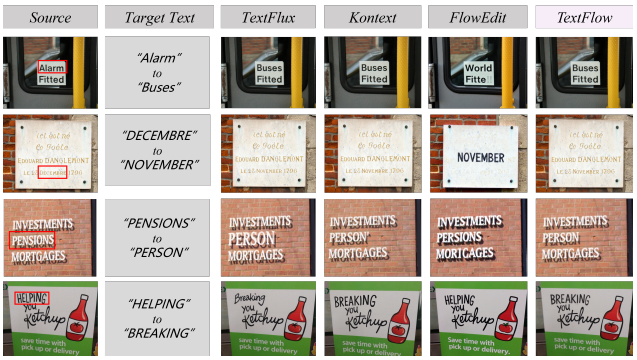

Figure 5. Qualitative comparison among different DiT-based methods on a full-size image.

4.4. Ablation Study

To comprehensively evaluate the contributions of different components in our proposed framework, we conduct systematic ablation studies across three key aspects: the FMS module for structural preservation, the AttnBoost mechanism for text rendering accuracy, and the optimization of inference configurations, including scheduler selection and step count. These experiments validate the necessity of each component and identify optimal parameter settings.

Table 2. Ablation of FMS modules for image quality.

FMS Module	SSIM ↑	PSNR ↑	MSE ↓	FID ↓
FlowEdit [20]	<u>87.60</u>	<u>20.89</u>	<u>1.16</u>	25.41
Ours w/o FMS	87.09	20.47	1.35	<u>16.69</u>
Ours w FMS	**89.04**	**22.42**	**0.97**	**13.52**

Table 2 presents the ablation results evaluating our proposed FMS module. Our full method with FMS achieves the best performance across all image quality metrics, with 89.04 SSIM, 22.42 PSNR, 0.97 MSE, and 13.52 FID.

Compared to FlowEdit [20], our method shows substantial improvements, increasing SSIM from 87.60 to 89.04 and PSNR from 20.89 to 22.42 while reducing FID from 25.41 to 13.52. Removing the FMS module causes significant degradation, with PSNR dropping by 1.95 and MSE increasing by 39.2%, confirming the critical importance of our trajectory correction. Although the ablated version maintains an FID advantage over FlowEdit [20], the comprehensive superiority of our full method demonstrates that

FMS effectively balances structural preservation with visual quality enhancement.

As demonstrated in Fig. 6 (a), the incorporation of FMS significantly enhances style consistency between the original and edited images while notably improving the preservation of fine-grained details.

Table 3. Ablation of AttnBoost considering text accuracy.

AttnBoost Module	ScenePair		ScenePair (Random)	
	ACC(%) ↑	NED ↑	ACC(%) ↑	NED ↑
FLUX-Kontext [22]	78.72	0.920	**76.63**	**0.916**
Ours w/o AttnBoost	20.35	0.420	18.84	0.391
Ours w AttnBoost	**79.80**	**0.931**	74.52	0.874

Table 3 presents that the AttnBoost module can significantly enhance textual accuracy. On the ScenePair dataset, our full model with AttnBoost achieves the best performance with 79.80% accuracy and 0.931 NED, outperforming both the FLUX-Kontext [22] baseline and the ablated version. Although FLUX-Kontext [22] performs best on the more challenging ScenePair Random dataset, our method remains competitive. Removing AttnBoost causes a dramatic performance drop, with accuracy decreasing by approximately 75% and NED by 55%, confirming its essential role in high-quality text rendering.

The Fig. 6 (b) reveals that AttnBoost substantially improves textual accuracy, with particularly notable enhancements observed in challenging cases involving long words and consecutive characters.

Table 4. Ablation of inference steps on ScenePair.

Steps	SSIM ↑	PSNR ↑	MSE ↓	FID ↓	ACC(%) ↑	NED ↑
24	86.80	20.21	1.43	16.94	77.97	0.925
30	87.12	19.86	1.46	23.1	79.90	**0.928**
42	88.04	22.21	0.97	52.8	79.40	0.926
50	**89.30**	**22.47**	0.91	13.53	**79.98**	0.914
70	87.01	21.02	**0.90**	**12.83**	79.88	0.914

Table 4 presents a comprehensive comparison of inference steps across both generative and render metrics. Our experiments demonstrate that 50 denoising steps achieve the optimal balance between generation quality and textual accuracy while maintaining computational efficiency.

In terms of image quality metrics, 50 steps yield the best overall performance with 89.30 SSIM, 22.47 PSNR, and 13.53 FID, while achieving a competitive MSE of 0.91. For textual accuracy, 50 steps produce the highest character accuracy of 79.98% with 0.914 NED. Although 70 steps achieve slightly better MSE and FID scores, the improvements are marginal while requiring significantly more computational resources.

The results indicate that 50 steps yield the most efficient operating point, delivering superior visual quality and text fidelity without the computational overhead associated with higher step counts. This balanced performance makes

Figure 6. Qualitative ablation studies validate the effectiveness of FMS in style preservation and demonstrate the significant improvement in text rendering accuracy achieved by AttnBoost.

50 steps the recommended setting for practical applications where both quality and efficiency are prioritized.

Table 5. Ablation of scheduler on the ScenePair dataset.

Scheduler	ACC(%) ↑	NED ↑
Ours w Euler	78.73	0.920
Ours w Overshoot [16]	**79.90**	**0.931**

Table 5 presents that the Overshoot scheduler consistently outperforms the Euler scheduler in text rendering accuracy. Our method with the Overshoot scheduler achieves superior performance, reaching 79.90% accuracy and 0.931 NED, compared to 78.73% accuracy and 0.920 NED with the Euler scheduler. This demonstrates that the Overshoot scheduler, which extends the denoising trajectory beyond conventional bounds, provides more precise control over text generation, thereby improving character accuracy and editing quality.

5. Conclusion and Limitation

We introduce TextFlow, a training-free framework for scene text editing that balances structural preservation with textual accuracy. It integrates two complementary components: FMS maintains structural consistency via trajectory guidance in early phases, while AttnBoost enables fine-grained text rendering in later phases. This integration establishes a new paradigm for phase-aware generative guidance. Extensive experiments demonstrate state-of-the-art performance in both image quality and text accuracy, delivering high-fidelity edits without task-specific training or large-scale paired datasets.

Despite these advances, certain limitations remain. The computational overhead of the underlying diffusion model limits real-time applicability, especially for high-resolution outputs. More notably, the framework struggles with multi-line text and complex layouts, where maintaining spatial and typographic consistency proves challenging.

Acknowledgement

This work is Funded by Basic Research Program of Jiangsu (BK20251441, BK20252040, BK20251414).

References

[1] Omri Avrahami, Or Patashnik, Ohad Fried, Egor Nemchinov, Kfir Aberman, Dani Lischinski, and Daniel Cohen-Or. Stable flow: Vital layers for training-free image editing. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 7877–7888, 2025. 3

[2] 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, 2019. 6

[3] Siyu Cao, Hangting Chen, Peng Chen, Yiji Cheng, Yutao Cui, Xinchi Deng, Ying Dong, Kipper Gong, Tianpeng Gu, Xiusen Gu, et al. Hunyuanimage 3.0 technical report. *arXiv preprint arXiv:2509.23951*, 2025. 2

[4] 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*, 36:63062–63074, 2023. 1, 2

[5] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser: Diffusion models as text painters, 2023. 2, 6, 7

[6] Jiuhai Chen, Le Xue, Zhiyang Xu, Xichen Pan, Shusheng Yang, Can Qin, An Yan, Honglu Zhou, Zeyuan Chen, Lifu Huang, Tianyi Zhou, Junnan Li, Silvio Savarese, Caiming Xiong, and Ran Xu. Blip3o-next: Next frontier of native image generation, 2025. 2

[7] Alloy Das, Sanket Biswas, Prasun Roy, Subhankar Ghosh, Umapada Pal, Michael Blumenstein, Josep Lladós, and Saumik Bhattacharya. Faster: A font-agnostic scene text editing and rendering framework. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1944–1954. IEEE, 2025. 1

[8] Nikai Du, Zhennan Chen, Shan Gao, Zhizhou Chen, Xi Chen, Zhengkai Jiang, Jian Yang, and Ying Tai. Textcrafter: Accurately rendering multiple texts in complex visual scenes. *arXiv preprint arXiv:2503.23461*, 2025. 3

[9] Yongkun Du, Miaomiao Zhao, Songlin Fan, Zhineng Chen, Caiyan Jia, and Yu-Gang Jiang. Mdiff4str: Mask diffusion model for scene text recognition, 2025. 1

[10] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 1, 2

[11] Hao Guo, Xugong Qin, Jun Jie Ou Yang, Peng Zhang, Gangyan Zeng, Yubo Li, and Hailun Lin. Towards natural language-based document image retrieval: New dataset and benchmark. In *CVPR*, pages 29722–29732, 2025. 1

[12] Youhui Guo, Yu Zhou, Xugong Qin, and Weiping Wang. Which and where to focus: a simple yet accurate framework for arbitrary-shaped nearby text detection in scene images. In *International Conference on Artificial Neural Networks*, pages 271–283. Springer, 2021.

[13] Youhui Guo, Yu Zhou, Xugong Qin, Enze Xie, and Weiping Wang. Units: Unsupervised intermediate training stage for scene text detection. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022. 1

[14] 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. 6

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[16] Xixi Hu, Keyang Xu, Bo Liu, Qiang Liu, and Hongliang Fei. Amo sampler: Enhancing text rendering with overshooting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13157–13166, 2025. 3, 6, 8, 1

[17] 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. *arXiv preprint arXiv:2304.05568*, 2023. 1, 2, 6, 7

[18] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1484–1493. IEEE, 2013. 5, 2

[19] 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(7):9122–9134, 2023. 1

[20] Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. *arXiv preprint arXiv:2412.08629*, 2024. 3, 6, 7, 2

[21] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 1, 2, 6, 7

[22] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. 2, 4, 6, 7, 8, 3

[23] Rui Lan, Yancheng Bai, Xu Duan, Mingxing Li, Dongyang Jin, Ryan Xu, Lei Sun, and Xiangxiang Chu. Flux-text: A simple and advanced diffusion transformer baseline for scene text editing. *arXiv preprint arXiv:2505.03329*, 2025. 1, 2, 3

[24] 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. 5, 2

[25] Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, and Jean Marc Ogier. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification - rrc-mlt. *IEEE*, 2017. 5, 2

[26] OpenAI. Gpt-4o system card, 2024. 2

[27] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 2

[28] Zhi Qiao, Xugong Qin, Yu Zhou, Fei Yang, and Weiping Wang. Gaussian constrained attention network for scene text recognition. In *ICPR*, pages 3328–3335. IEEE, 2021. 1

[29] Xugong Qin, Yu Zhou, Dongbao Yang, and Weiping Wang. Curved text detection in natural scene images with semi- and weakly-supervised learning. In *ICDAR*, pages 559–564. IEEE, 2019. 1

[30] Xugong Qin, Yu Zhou, Youhui Guo, Dayan Wu, Zhihong Tian, Ning Jiang, Hongbin Wang, and Weiping Wang. Mask is all you need: Rethinking mask r-cnn for dense and arbitrary-shaped scene text detection. In *ACM Multimedia*, pages 414–423, 2021.

[31] Xugong Qin, Yu Zhou, Youhui Guo, Dayan Wu, and Weiping Wang. Fc2rn: a fully convolutional corner refinement network for accurate multi-oriented scene text detection. In *ICASSP*, pages 4350–4354. IEEE, 2021.

[32] Xugong Qin, Pengyuan Lyu, Chengquan Zhang, Yu Zhou, Kun Yao, Peng Zhang, Hailun Lin, and Weiping Wang. Towards robust real-time scene text detection: From semantic to instance representation learning. In *ACM Multimedia*, pages 2025–2034, 2023.

[33] Xugong Qin, Jiuqiang Tian, Jiayi Sheng, Tiantian Xia, Yuyi Wang, Chengrui Li, and Gangyan Zeng. Towards fine-grained document tampering detection: New dataset and benchmark. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 3–19. Springer, 2025.

[34] Xugong Qin, Peng Zhang, Jun Jie Ou Yang, Gangyan Zeng, Yubo Li, Yuanyuan Wang, Wanqian Zhang, and Pengwen Dai. Clip is almost all you need: Towards parameter-efficient scene text retrieval without ocr. In *CVPR*, pages 24873–24883, 2025. 1

[35] 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. 1

[36] 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. 1

[37] Divya Saxena and Jiannong Cao. Generative adversarial networks (gans) challenges, solutions, and future directions. *ACM Computing Surveys (CSUR)*, 54(3):1–42, 2021. 1

[38] Meituan LongCat Team, Hanghang Ma, Haoxian Tan, Jiale Huang, Junqiang Wu, Jun-Yan He, Lishuai Gao, Songlin Xiao, Xiaoming Wei, Xiaoqi Ma, Xunliang Cai, Yayong Guan, and Jie Hu. Longcat-image technical report. *arXiv preprint arXiv:2512.07584*, 2025. 3

[39] Xunquan Tong, Pengwen Dai, Xugong Qin, Rui Wang, and Wenqi Ren. Granularity-aware single-point scene text spotting with sequential recurrence self-attention. *TCSVT*, 2024. 1

[40] Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Geng, and Xuansong Xie. Anytext: Multilingual visual text generation and editing. *arXiv preprint arXiv:2311.03054*, 2023. 1, 2, 6, 7

[41] Yuxiang Tuo, Yifeng Geng, and Liefeng Bo. Anytext2: Visual text generation and editing with customizable attributes. *arXiv preprint arXiv:2411.15245*, 2024. 2

[42] Changshuo Wang, Lei Wu, Xu Chen, Xiang Li, Lei Meng, and Xiangxu Meng. Letter embedding guidance diffusion model for scene text editing. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 588–593. IEEE, 2023. 1

[43] Jiangshan Wang, Junfu Pu, Zhongang Qi, Jiayi Guo, Yue Ma, Nisha Huang, Yuxin Chen, Xiu Li, and Ying Shan. Taming rectified flow for inversion and editing. *arXiv preprint arXiv:2411.04746*, 2024. 3

[44] Tengfei Wang, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen. Pretraining is all you need for image-to-image translation, 2022. 1

[45] Tong Wang, Ting Liu, Xiaochao Qu, Chengjing Wu, Luoqi Liu, and Xiaolin Hu. Glyphmastero: A glyph encoder for high-fidelity scene text editing, 2025. 2

[46] Yibin Wang, Weizhong Zhang, Honghui Xu, and Cheng Jin. Dreamtext: High fidelity scene text synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28555–28563, 2025. 2

[47] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. 2, 6, 7, 3

[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. 1

[49] Weiyan Xie, Han Gao, Didan Deng, Kaican Li, April Hua Liu, Yongxiang Huang, and Nevin L. Zhang. Cannyedit: Selective canny control and dual-prompt guidance for training-free image editing. *arXiv preprint arXiv:2508.06937*, 2025. 3

[50] 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. 1, 2, 6, 7, 3

[51] 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. 1

[52] Yunjie Yu, Jingchen Wu, Junchen Zhu, Chunze Lin, and Guibin Chen. Skyreels-text: Fine-grained font-controllable text editing for poster design, 2025. 1

[53] Gangyan Zeng, Yuan Zhang, Jin Wei, Dongbao Yang, Peng Zhang, Yiwen Gao, Xugong Qin, and Yu Zhou. Focus, distinguish, and prompt: Unleashing clip for efficient and flexible scene text retrieval. In *ACM Multimedia*, pages 2525–2534, 2024. 1

[54] 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, 5, 3

[55] Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. In-context edit: Enabling instructional image editing with in-context generation in large scale diffusion transformer. *arXiv preprint arXiv:2504.20690*, 2025. 3

[56] Yiming Zhao and Zhouhui Lian. Udifftext: A unified framework for high-quality text synthesis in arbitrary images via character-aware diffusion models. In *European conference on computer vision*, pages 217–233. Springer, 2024. 2

[57] Yiming Zhao, Yuanpeng Gao, Yuxuan Luo, Jiwei Duan, Shisong Lin, Longfei Xiong, and Zhouhui Lian. Utdesign: A unified framework for stylized text editing and generation in graphic design images. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, page 1–11. ACM, 2025. 1

[58] Candi Zheng, Yuan Lan, and Yang Wang. Lanpaint: Training-free diffusion inpainting with asymptotically exact and fast conditional sampling, 2025. 3

[59] 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. 1

[60] Tianrui Zhu, Shiyi Zhang, Jiawei Shao, and Yansong Tang. Kv-edit: Training-free image editing for precise background preservation. *arXiv preprint arXiv:2502.17363*, 2025. 3