Self-Prompting Diffusion Transformer for Open-Vocabulary Scene Text Editing via In-Context Learning

Hongxi Li [1 2] **Tong Wang** [1] **Chengjing Wu** [1] **Tianbao Liu** [1] **Jiangtao Yao** [1] **Xiaochao Qu** [1] **Xinxiao Wu** [2]
Luoqi Liu [1] **Ting Liu** [1]

Abstract

Scene text editing aims to modify text in a target region of an image while preserving surrounding background style and texture. Existing methods rely solely on image background information while neglecting the visual details of target regions, which discards stylistic features in the original text and essentially degrades the task to text rendering. Moreover, the conditions imposed by pre-trained glyph encoder limit the scope of editable text. To address these issues, this paper proposes a self-prompting scene text editing method that constructs style and glyph prompts directly from the original image, without introducing additional style or glyph encoders. We employ a two-stage training strategy: the diffusion transformer is first trained on large-scale self-supervised data and then refined using a small set of paired images. By leveraging the in-context learning capability of the Multi-Modal Diffusion Transformer (MM-DiT), it achieves open-vocabulary and style-consistent text editing. Experimental results on various languages demonstrate that our method achieves the state-of-the-art performance in both text accuracy and style consistency. Our project page: hongxiii.github.io/mstedit.

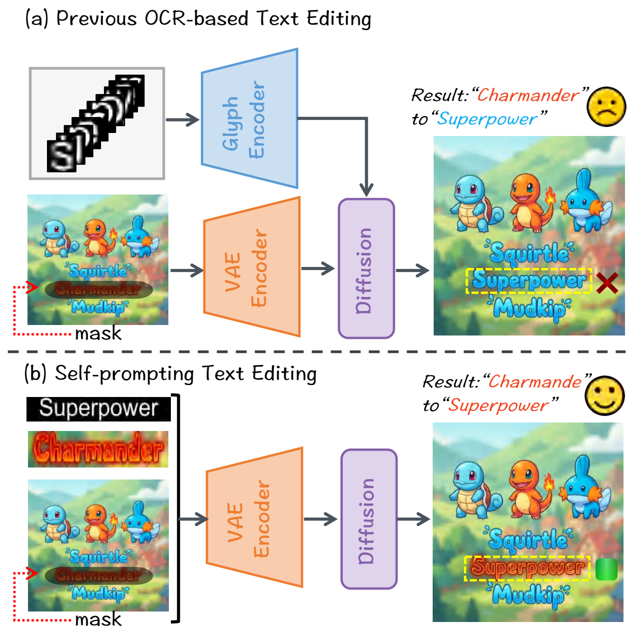

Figure 1. Comparison of previous OCR-based text edit and our proposed self-prompting text edit.

1. Introduction

Scene text editing is a specialized yet fundamental image editing task that aims to modify textual content in natural scene images while strictly preserving glyph correctness and visual consistency with the surrounding context. By formulating text modification as targeted content generation within localized text regions, scene text editing naturally aligns with image inpainting, while imposing additional constraints on semantic fidelity, typographic structure, and cross-modal consistency. Driven by recent advances in diffusion-based image inpainting, recent methods (Tuo et al., 2023; 2024; Zeng et al., 2024; Wang et al., 2025a;b) have made notable progress in scene text editing. However, these methods inherit the intrinsic limitations of inpainting-based formulations: they struggle to capture fine-grained glyph structures—particularly for logographic scripts and rare languages—and inevitably discard visual information within the original text regions. As a result, edited outputs often suffer from character distortion and degraded consistency in style and texture with respect to the source text.

The main challenge in scene text editing lies in capturing the glyph features of target text. An intuitive approach is to utilize a pre-trained optical character recognition (OCR) (Du et al., 2020) feature extractor as the glyph encoder. However,

[1]MT Lab, Meitu Inc., Beijing, China [2]School of Computer Science & Technology, Beijing Institute of Technology, Beijing, China. Correspondence to: Ting Liu <lt@meitu.com>, Luoqi Liu <llq5@meitu.com>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

as OCR systems are inherently designed as classifiers, their fixed vocabularies fundamentally limit the scope of scene text editing, whereas a glyph encoder trained from scratch relies on massive image data. In addition, existing methods take the design paradigm of image inpainting for text editing, which discards the original information of the target region. As a consequence, the generated text often fails to preserve the pre-editing style and instead borrows stylistic cues from surrounding regions. As illustrated in Figure 1(a), although previous methods have modified the red word ”Charmander” to the target word ”Superpower”, the target text retains the same blue font as the surrounding text, thereby losing its original stylistic texture.

To address these issues, this paper proposes a self-prompting open-vocabulary text editing method. We capture glyph features at the stroke level rather than the character level, without the need for additional encoders. We leverage the in-context learning capability of the Multi-Modal Diffusion Transformer (MMDiT) (Labs, 2024) to to enhance the model's generalization performance. This design enables the model to adapt to complex real-world scenarios and support diverse languages with only a small-scale paired dataset, while simultaneously ensuring strict style consistency between the original and edited text. As demonstrated in Figure 1(b), our method successfully modifies the red word ”Charmander” to the red word ”Superpower”. Notably, the edited text retains the original color, font, and texture of the target region, even though the surrounding text in the background remains blue.

Specifically, for input prompt construction, we generate a high-fidelity glyph map by rendering the target text, which serves as the glyph prompt to guide text content generation. Concurrently, we extract the original pixel information from the target text region in the input image to form a style prompt, which encodes the unique color, texture, and font style of the original text. These two prompts are then concatenated with the full input image to form a multi-modal input tensor for the MMDiT backbone. Second, we freeze the model's encoder and decoder components and exclusively train the backbone using large-scale self-supervised image-text datasets. This step equips the model with fundamental text inpainting capabilities without overfitting to specific text styles. Finally, we utilize a mask-free image editing tool to collect and filter high-quality paired image datasets, where each pair consists of an original image and its corresponding style-consistent edited version. These curated datasets are then used for the cooldown training, during which the model learns to align the generated text with the original style of the target region.

The main contributions of our proposed self-prompting scene text edit can be summarized as follow:

- We achieve open-vocabulary text editing by in-context learning to capture stroke-level features from rendered glyph images, rather than character-level features.
- We enhance pre- and post-editing style consistency of the target text via cooldown training on a limited amount of paired data, without introducing additional style encoders.
- Extensive experiments on the AnyWord-3M and MST-Edit datasets demonstrate that our method outperforms existing methods in text accuracy and style fidelity across 13 evaluated languages.

2. Related Work

2.1. Image Inpainting

Image Inpainting aims to fill missing image regions with visually coherent and semantically plausible content while remaining consistent with surrounding context. Early methods rely on patch-based propagation and GAN-based encoder–decoder architectures with structural priors such as partial or gated convolutions and contextual attention (Liu et al., 2018; Yu et al., 2018; 2019). More recent advances are dominated by diffusion-based approaches, which can be categorized into sampling-based methods that modify the denoising process in a training-free manner (Avrahami et al., 2022; Lugmayr et al., 2022), and fine-tuning-based methods that explicitly encode masks and masked images to improve content and shape awareness (Rombach et al., 2022; Manukyan et al., 2023; Zhuang et al., 2024). To balance performance and generality, recent plug-and-play designs decouple masked image conditioning from generation, enabling effective inpainting without retraining entire diffusion backbones (Zhang et al., 2023; Ju et al., 2024).

Image inpainting serves as the foundation of scene text editing, where accurate background restoration is crucial for consistent text synthesis.

2.2. Scene Text Editing

Scene text editing extends image inpainting by jointly requiring background restoration and geometry- and style-consistent text synthesis. Early methods (Roy et al., 2020; Yang et al., 2020; Qu et al., 2023) adopt multi-stage GAN-based pipelines with explicit geometric or stroke priors, offering controllable edits but suffering from poor generalization and error accumulation in complex scenes. Recent methods (Tuo et al., 2023; 2024; Wang et al., 2025b)employ pre-trained OCR models as glyph encoders to capture character-level structures. While this design improves glyph awareness, it inherently restricts generalization to characters outside the fixed OCR vocabulary. In contrast, Flux-Text (Lan et al., 2025) and TextFlux (Xie et al., 2025) remove glyph encoders and instead render glyphs directly within masked regions to extract visual features; however, the spatial extent of the target region can hinder the accurate capture fine-grained glyph details. Despite their differences, these methods largely adhere to an image inpainting formulation that discards the original text region, leading to the loss of pre-existing style information and degraded stylistic consistency during editing. TextCtrl (Zeng et al., 2024) employs a text style encoder to capture color, font, texture, and background features from the original text. However, the representation mismatch between the glyph-structure encoder and the VAE image latent space restricts the method's scalability beyond isolated text regions.

Differing from existing methods, our method directly constructs style and glyph prompts from the original image without additional encoders, thereby enabling open-vocabulary and style-consistent text editing.

3. Method

3.1. Preliminary

Our method is built upon *FLUX-Fill* (Labs, 2024), an inpainting-oriented variant of MMDiT, which formulates image editing as a conditional rectified flow process in the latent space and employs a transformer-based architecture to jointly reason over visual content, spatial structure, and textual semantics under multimodal conditions.

Masked Image Construction. Given an input image $I \in \mathbb{R}^{H\times W\times 3}$ and a binary mask $M \in \{0,1\}^{H\times W}$ indicating the target text region, FLUX-Fill first constructs a masked image

$$I_m = I \odot (1 - M), \tag{1}$$

where $\odot$ denotes element-wise multiplication. The masked image I_m preserves the visual context outside the editable region while removing the original content within the target area. Both the original image I and the masked image I_m are encoded into latent representations using a frozen VAE encoder, producing visual tokens that serve as the input to the diffusion transformer. The binary mask M is also embedded and provided to the model as an explicit spatial prior, enabling region-aware generation during denoising.

Text Prompt Encoding. FLUX-Fill employs two complementary text encoders to extract semantic guidance from textual prompts. Specifically, a T5 encoder is used to process the full natural-language prompt, capturing high-level semantic intent and contextual information. The resulting embeddings are injected into the diffusion transformer through cross-attention layers, guiding global content generation and scene-level consistency. In parallel, a CLIP text encoder is used to encode concise textual descriptors that emphasize visual alignment, such as object names or short phrases. CLIP embeddings primarily serve to enhance vision–language alignment and stabilize the correspondence between generated content and visual context.

Dual-Stream and Single-Stream Transformer. The core of FLUX-Fill is a hybrid transformer design that alternates between dual-stream and single-stream blocks. In the dual-stream transformer blocks, visual tokens and text tokens are processed in separate streams with independent self-attention operations. Cross-attention is then applied to enable information exchange between modalities, allowing the model to align textual semantics with spatially grounded visual representations while maintaining modality-specific feature structures.

After cross-modal interaction, the architecture transitions to single-stream transformer blocks, where visual and text tokens are concatenated into a unified token sequence and processed jointly.

Rectified Flow Objective. Let z_0 denote the clean latent representation obtained from the VAE encoder and $z_1 \sim \mathcal{N}(0, I)$ be Gaussian noise. A noisy latent z_t is constructed via rectified flow interpolation at timestep t, and the model is trained to predict the velocity field connecting z_t and z_0:

$$\mathcal{L}_{\text{RF}} = \mathbb{E}_{t,z_0,z_1}\left[\left\|\hat{v}_\theta(z_t, t, c) - (z_1 - z_0)\right\|_2^2\right], \tag{2}$$

where c denotes the multimodal conditioning inputs, including visual tokens, text embeddings, and spatial mask information.

3.2. Overall Architecture

Our method explicitly disentangles text style and text content through dedicated visual and textual prompts, while fully leveraging the in-context learning capability of the MMDiT backbone. An overview of the proposed architecture is shown in Figure 2.

Style Prompt Construction. To preserve the visual appearance of the original text, we construct a *style prompt* from both visual and textual perspectives.

Given an input image $I \in \mathbb{R}^{H\times W\times 3}$ and a binary mask $M \in \{0,1\}^{H\times W}$ indicating the target text region, we compute the maximal enclosing bounding rectangle of the masked area and crop the corresponding region from I to obtain the visual style prompt I_s. This cropped patch encodes region-specific appearance information, such as color, texture, font characteristics, and local illumination, and serves as a visual reference for style preservation.

For style text prompt, we encode the input text description using the CLIP text encoder. Together, the visual and textual style prompts enable the model to infer and preserve the intrinsic style of the target text region without introducing additional style-specific encoders.

Glyph Prompt Construction. To guide the generation

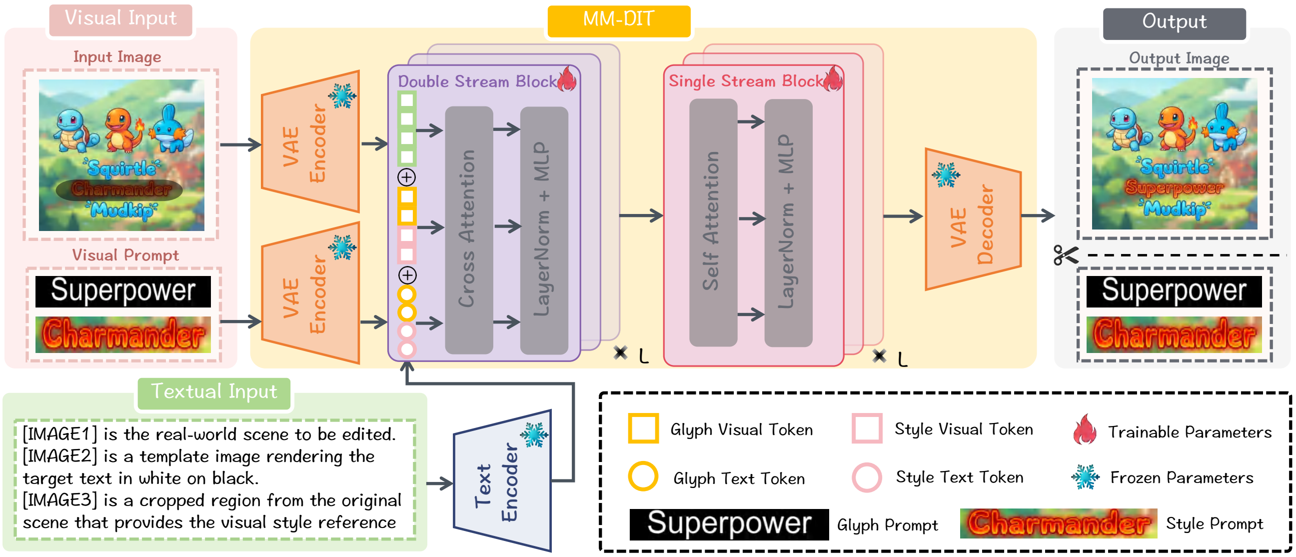

Figure 2. Overview of our proposed method.

of target text content, we construct a *glyph prompt* that explicitly represents the desired textual structure.

Specifically, the target text is rendered into a single-line glyph image using the Pillow library, producing a white-on-black glyph map I_g. This high-contrast representation preserves fine-grained stroke-level geometry and provides explicit structural guidance for complex glyphs across different languages and scripts, reducing the burden on the diffusion model to learn character structures from scratch.

For glyph text prompt, we encode the target text string using the T5 text encoder. The resulting embeddings capture high-level semantic and syntactic information of the target text and are injected into the MMDiT backbone through cross-attention, guiding content-aware generation.

Denoising Process. The visual glyph prompt I_g, visual style prompt I_s, and masked image I_m are concatenated along the channel dimension to form the composite visual input

$$I_{\text{input}} = \text{Concat}(I_g, I_s, I_m)$$

The composite input is encoded by a frozen VAE encoder $\mathcal{E}(\cdot)$ to obtain the latent representation z_0, which is processed by the MMDiT backbone together with the textual glyph and style embeddings.

After the denoising process, the predicted latent representation is decoded by the VAE decoder $\mathcal{D}(\cdot)$ to produce an edited image. Since the masked image is used only for conditioning, the final output is obtained by cropping the decoded result to the spatial region corresponding to the target text area.

3.3. Cooldown Training

Existing OCR-based datasets discard original text appearance by masking target regions, providing only self-supervised rendering signals and limiting style preservation. To enable style-consistent text editing under limited paired data, we adopt a two-stage cooldown training strategy.

We first construct a paired image dataset tailored for style-aware learning. Specifically, we leverage an instruction-based image editing model (*Nano Banana Pro* [1]) to generate edited images conditioned on explicit editing instructions. Each data pair consists of an original image and a corresponding edited image, in which only the target text region is modified while all other regions are strictly preserved. To ensure data reliability, we manually filter the generated pairs and retain only samples that satisfy the following criteria:

- Non-target regions remain unchanged.
- The generated text content is semantically accurate.
- The edited text preserves consistent style, color, and texture with respect to the original text.

Self-supervised Pretraining. We first pretrain the model on the AnyWord-3M dataset, which provides large-scale self-supervised data for multilingual scene text rendering. During this stage, the optimization objective follows Eq. 2, where the conditioning includes multimodal visual and textual inputs.

Cooldown Training. Inspired by MECO (Gao et al., 2025), we treat the original text region in the input image as meta-information during training. We continue training from the

[1] https://nanobanana.im/nano-banana-pro

pretrained checkpoint using a curated paired dataset consisting of 4,000 manually filtered image pairs. Each pair contains an original image and a corresponding edited image with style-consistent text replacement. Detailed construction process are provided in Appendix C. The comparison between data used by pre-training and cooldown stage is shown in Figure 3

To mitigate the degenerate optimization where original text region features unduly dominate the target region (leading the model to replicate both original style and glyphs), we design a target-region-oriented training objective for the cooldown stage. This objective restricts the learning signal to localized text transformations, prompting the model to decouple style preservation from content regeneration in optimization.

Let z_0^{src} and z_0^{tgt} denote the latent representations of the source image and its corresponding edited image, respectively. We construct an interpolated latent variable

$$z_t = (1 - \sigma_t)\, z_0^{\mathrm{src}} + \sigma_t\, z_0^{\mathrm{tgt}}, \tag{3}$$

where $\sigma_t \in [0, 1]$ is a time-dependent interpolation coefficient. The model is trained to predict the velocity field that connects the source and target latents by minimizing

$$\mathcal{L}_{\mathrm{CD}} = \mathbb{E}_t \left[\left\| \hat{v}_\theta(z_t, t, c) - \left(z_0^{\mathrm{tgt}} - z_0^{\mathrm{src}}\right) \right\|_2^2 \right], \tag{4}$$

where the conditioning c includes the original text region as meta-information through the style prompt.

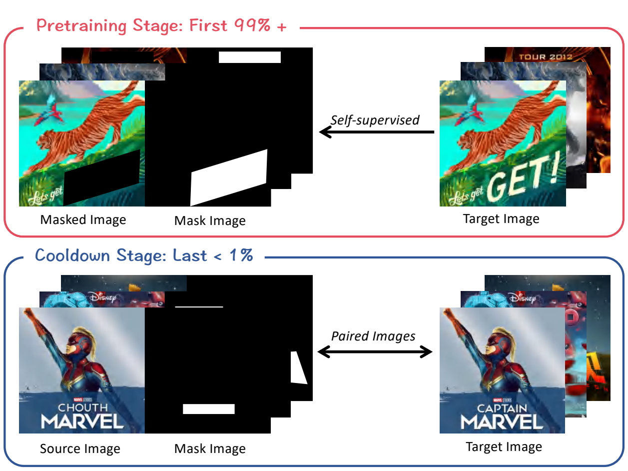

Figure 3. A comparison between data used by standard pre-training and cooldown training.

4. Experiment

4.1. Experimental Setup

Datasets. We adopt AnyWord-3M (Tuo et al., 2023) as the large-scale benchmark dataset. Its training set contains 1.6M Chinese images, 1.39M English images, and 10K images spanning Japanese, Korean, Arabic, Bengali, and Hindi. The test split includes 1,000 Chinese and 1,000 English images, and is denoted as AnyText-benchmark.

Table 1. Composition of the MST-Edit dataset.

Source Dataset	Language	# Images
ICDAR-19	Arabic	1,000
	French	1,000
	German	1,000
	Korean	1,000
	Japanese	1,000
	Italian	1,000
	Bengali	1,000
	Hindi	1,000
RusTitW	Russian	3,795
ThaiOCRBench	Thai	2,808
Swahili-STR	Swahili	985

We further construct a Multi-lingual Scene Text Editing Dataset (MST-Edit) by aggregating multiple publicly available multilingual datasets, including ICDAR-19 (Nayef et al., 2019), RusTitW (Markov et al., 2023), ThaiOCR-Bench (Nonesung et al., 2025), and Swahili-STR (Douamba et al., 2024). Detailed statistics of the dataset composition are summarized in Table 1. We randomly split 20% of MST-Edit for testing, with the remainder used for training.

Evaluation metrics. For textual accuracy, Sentence Accuracy (Seq. ACC) quantifies the correctness of generated text at the sentence level, and Normalized Edit Distance (NED) measures character-level similarity between the generated and target text. For visual fidelity, Fréchet Inception Distance (FID) (Seitzer, 2020) quantifies the distribution alignment of generated and real images in the Inception-v3 feature space, and Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018) measures the L2 distances between perceptual VGG-based deep features.

Implementation Details. All experiments are conducted on a cluster with 8 NVIDIA A100 GPUs. Our model is initialized from FLUX.1-Fill-Dev[2]. During training, we freeze the VAE, CLIP text encoder, and T5 text encoder, and update only the transformer parameters. We follow the default FLUX configuration, using a fixed guidance scale of 30 and 30 sampling steps for both training and inference.

Training is conducted in two stages. We first train the model on AnyWord-3M for one epoch, followed by a 10-epoch cooldown phase on the paired image dataset. Multilingual data are jointly mixed without language-specific scheduling. We use the AdamW optimizer with a constant learning rate

[2] `https://huggingface.co/black-forest-labs/FLUX.1-Fill-dev`

Table 2. Quantitative results on AnyText-benchmark. **Bold** indicates the best result and <u>underline</u> indicates the second best.

Method	English				Chinese			
	Sen. Acc(↑)	NED(↑)	FID(↓)	LPIPS(↓)	Sen. Acc(↑)	NED(↑)	FID(↓)	LPIPS(↓)
TextDiffuser (Chen et al., 2023)	0.5176	0.7618	29.76	0.1564	0.0559	0.1218	34.19	0.1252
AnyText (Tuo et al., 2023)	0.6843	0.8588	21.59	0.1106	0.6476	0.8210	20.01	0.0943
TextCtrl (Zeng et al., 2024)	0.5853	0.8146	35.73	0.1978	0.3580	0.6084	49.79	0.2298
AnyText2 (Tuo et al., 2024)	0.7915	0.9100	29.76	0.1734	0.7022	0.8420	26.52	0.1444
GlyphMastero (Wang et al., 2025a)	0.8170	-	-	-	0.7301	-	-	-
FIUX-Text (Lan et al., 2025)	0.8175	0.9193	<u>12.35</u>	<u>0.0674</u>	0.7213	0.8555	<u>12.41</u>	<u>0.0487</u>
TextFlux (Xie et al., 2025)	<u>0.8231</u>	<u>0.9235</u>	13.42	0.0721	<u>0.7289</u>	<u>0.8612</u>	13.67	0.0524
Our method	**0.8857**	**0.9568**	**7.62**	**0.0365**	**0.8249**	**0.9147**	**7.95**	**0.0268**

of 2×10^{-5}, bf16 mixed precision, and 8-bit optimizer states. The per-GPU batch size is 1 with gradient accumulation over 8 steps, resulting in an effective batch size of 64, and all experiments are conducted with a fixed random seed of 42.

4.2. Quantitative Result

We perform quantitative experiments under both bilingual (Chinese/English) and multilingual (non-Chinese/English) settings. For Chinese and English, we evaluate our method against state-of-the-art methods (Chen et al., 2023; Tuo et al., 2023; 2024; Zeng et al., 2024; Wang et al., 2025a; Xie et al., 2025; Lan et al., 2025) on the AnyText-benchmark.

For non-Chinese/English languages, we re-implement two OCR-free baselines (Xie et al., 2025; Lan et al., 2025) and compare them with our method on the MST-Edit dataset, which covers the remaining 11 languages.

Table 2 reports the quantitative results on the AnyText benchmark. Based on these results, we make the following observations:

(1) **Overall performance.** Our method achieves the best performance across all metrics on both the English and Chinese subsets, consistently outperforming existing methods.

(2) **Text accuracy.** Compared with the second-best method, TextFlux, our approach improves sentence-level accuracy from 0.8231 to 0.8857 on English and from 0.7289 to 0.8249 on Chinese. The NED score is also increased from 0.9235 to 0.9568 and from 0.8612 to 0.9147, respectively, indicating improved preservation of fine-grained glyph structures.

(3) **Image quality.** More significant gains are observed on image quality metrics. Our method reduces FID to 7.62/7.95 and LPIPS to 0.0365/0.0268 on English/Chinese, surpassing the strongest baselines and demonstrating superior visual fidelity and stylistic consistency.

The experimental results on MST-Edit are illustrated in Figure 4. Overall, our method consistently outperforms the two OCR-free baselines across all evaluated languages, demonstrating a clear method-level advantage. From a cross-

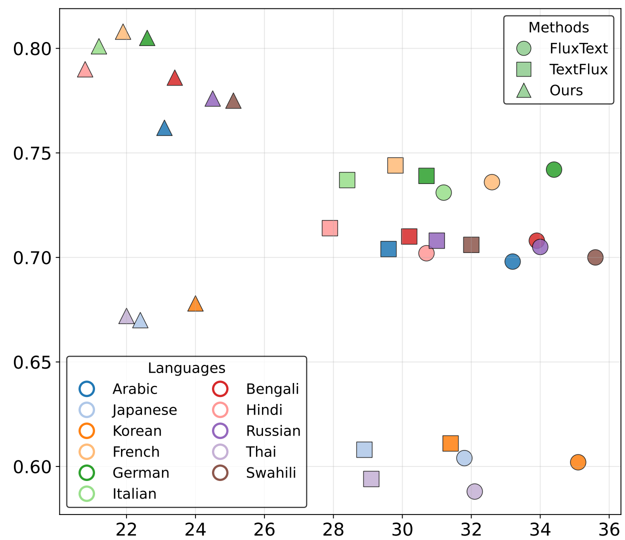

Figure 4. Quantitative comparison of OCR-free methods on MST-Edit, with Seq. ACC on the Y-axis and FID on the X-axis.

lingual perspective, languages belonging to the Latin script family (e.g., French, German, and Italian) generally achieve better performance, which can be attributed to their shared alphabetic structures and similar stroke patterns. In contrast, languages with more complex glyph compositions and diverse stroke layouts, such as Thai, tend to exhibit relatively lower performance. These results suggest that character structural complexity plays an important role in multilingual scene text editing, while our method remains robust across diverse writing systems. Complete results on MST-Edit are provided in the Appendix A.

4.3. Ablation Study

We conduct ablation studies to analyze the effect of style prompts and to investigate potential cross-lingual negative interference in multilingual scene text editing.

To analyze the impact of style prompts, we conduct an ablation study by removing visual style prompts from the image input and textual style prompts from the text input, while

Table 3. Ablation study on the effect of the cooldown stage.

	Metric	w/o Cooldown	w/ Cooldown
English	Sen. Acc (↑)	0.8738	**0.8857**
	NED (↑)	0.9470	**0.9568**
	FID (↓)	11.56	**7.62**
	LPIPS (↓)	0.0608	**0.0365**
Chinese	Sen. Acc (↑)	0.8125	**0.8249**
	NED (↑)	0.8906	**0.9147**
	FID (↓)	12.01	**7.95**
	LPIPS (↓)	0.0425	**0.0268**

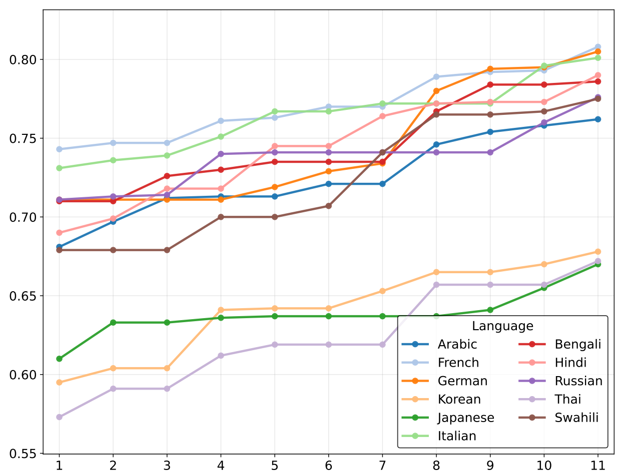

Figure 5. Effect of the number of introduced languages on Seq. ACC of initially introduced languages.

Figure 6. Comparison of scene text edit results with and without style prompts.

disabling the cooldown training described in Section 3.3.

Results on the AnyText benchmark are reported in Table 3. With the cooldown stage enabled, FID is reduced from 11.56 to 7.62 on English and from 12.01 to 7.95 on Chinese, while LPIPS decreases from 0.0608 to 0.0365 and from 0.0425 to 0.0268, respectively. In contrast, improvements in text accuracy are relatively modest. These results indicate that style prompts, reinforced by the cooldown stage, primarily improve image quality, while offering complementary gains in text accuracy.

We investigate the impact of progressively incorporating additional language data on earlier-introduced languages using a cyclic training protocol, in which languages are sequentially added in the order *Arabic, English, French, Chinese, German, Korean, Japanese, Italian, Bengali, Hindi, Russian, Thai, Swahili*.

Empirically, Seq. ACC remains stable and exhibits a consistent upward trend as additional languages are introduced, as illustrated in Figure 5. This behavior is consistent with our design principle of learning stroke-level visual primitives rather than language- and character-specific representations: shared low-level stroke structures across writing systems are reinforced by multilingual exposure, leading to improved generalization. Complete numerical results are provided in the Appendix A.

Figure 6 presents qualitative image editing results with and without style prompts. It can be observed that both settings accurately generate the target text without erroneous strokes. However, in the left example, when the style prompt is removed, the target text “HOT SUMMER” is rendered using a font style that does not appear in the original image, leading to noticeable stylistic inconsistency. In the right example, although the result without style prompts renders the target text “意识” using a font consistent with the surrounding text, it fails to preserve the original text background. In contrast, the setting with style prompts produces target text that more faithfully restores both the original font style and background, resulting in higher visual fidelity.

4.4. Qualitative Result

We conduct a qualitative comparison with representative multilingual scene text editing methods, including AnyText2 (Tuo et al., 2024), TextFlux (Xie et al., 2025), and

Figure 7. Qualitative results across Chinese, English, Korean, Japanese, Thai, and Russian.

FluxText (Labs, 2024).

As shown in Figure 7, our method consistently generates accurate target text across different scripts, preserving correct character structures and visual consistency. More specifically, results on Russian, Japanese, and Thai demonstrate the effectiveness of our method on long-tail languages with complex and diverse writing systems. In particular, our approach better preserves the original text style in the "战友定制" example, and remains robust in the "GOOD DOG!" case, where the masked region extends beyond the target text, introducing minimal unintended modifications to non-target regions. These observations indicate that our method generalizes well across languages while enabling precise and controlled text editing under challenging conditions.

5. Conclusion

This paper presents a self-prompting diffusion transformer framework for open-vocabulary scene text editing that preserves both textual correctness and visual style consistency by exploiting in-context information from the original image. By directly constructing glyph and style prompts from the input image, the proposed approach enables coherent text generation that remains faithful to the surrounding font appearance, layout, and visual context, without introducing additional glyph or style encoders. Extensive experiments on large-scale multilingual benchmarks demonstrate that our approach achieves strong performance across diverse languages and scripts, while maintaining robustness under progressive language expansion without negative transfer.

Overall, this paper provides a unified and scalable solution for multilingual scene text editing, and establishes a foundation for extending style-consistent editing to more complex and diverse real-world scenarios.

Impact Statement

This work studies scene text editing, which aims to modify textual content in images while preserving visual style and background consistency. The proposed method advances the flexibility and robustness of text editing across languages and scripts, which may benefit applications such as graphic design, content creation, and multilingual visual communication.

At the same time, like other image editing and generative techniques, scene text editing could be misused to alter visual content in misleading or deceptive ways. This work is intended for research purposes, and we encourage responsible use in accordance with existing ethical guidelines for generative models and visual media editing.

References

Avrahami, O., Lischinski, D., and Fried, O. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18208–18218, 2022.

Chen, J., Huang, Y., Lv, T., Cui, L., Chen, Q., and Wei, F. Textdiffuser: Diffusion models as text painters. *Advances in Neural Information Processing Systems*, 36: 9353–9387, 2023.

Douamba, F. W., Song, J., Fu, L., Liu, Y., and Bai, X. The first swahili language scene text detection and recognition dataset. In *International Conference on Document Analysis and Recognition*, pp. 215–226. Springer, 2024.

Du, Y., Li, C., Guo, R., Yin, X., Liu, W., Zhou, J., Bai, Y., Yu, Z., Yang, Y., Dang, Q., et al. Pp-ocr: A practical ultra lightweight ocr system. *arXiv preprint arXiv:2009.09941*, 2020.

Gao, T., Wettig, A., He, L., Dong, Y., Malladi, S., and Chen, D. Metadata conditioning accelerates language model pre-training. *arXiv preprint arXiv:2501.01956*, 2025.

Ju, X., Liu, X., Wang, X., Bian, Y., Shan, Y., and Xu, Q. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In *European Conference on Computer Vision*, pp. 150–168. Springer, 2024.

Labs, B. F. Flux. https://github.com/black-forest-labs/flux, 2024.

Lan, R., Bai, Y., Duan, X., Li, M., Jin, D., Xu, R., Sun, L., and Chu, X. Flux-text: A simple and advanced diffusion transformer baseline for scene text editing. *arXiv preprint arXiv:2505.03329*, 2025.

Liu, G., Reda, F. A., Shih, K. J., Wang, T.-C., Tao, A., and Catanzaro, B. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 85–100, 2018.

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., and Van Gool, L. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11461–11471, 2022.

Manukyan, H., Sargsyan, A., Atanyan, B., Wang, Z., Navasardyan, S., and Shi, H. Hd-painter: high-resolution and prompt-faithful text-guided image inpainting with diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2023.

Markov, I., Nesteruk, S., Kuznetsov, A., and Dimitrov, D. Rustitw: Russian language text dataset for visual text in-the-wild recognition. *arXiv preprint arXiv:2303.16531*, 2023.

Nayef, N., Patel, Y., Busta, M., Chowdhury, P. N., Karatzas, D., Khlif, W., Matas, J., Pal, U., Burie, J.-C., Liu, C.-l., et al. Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019. In *2019 International conference on document analysis and recognition (ICDAR)*, pp. 1582–1587. IEEE, 2019.

Nonesung, S., Jaknamon, T., Chaiophat, S., Nitarach, N., Wittayasakpan, C., Sirichotedumrong, W., Na-Thalang, A., and Pipatanakul, K. Thaiocrbench: A task-diverse benchmark for vision-language understanding in thai. *arXiv preprint arXiv:2511.04479*, 2025.

Qu, Y., Tan, Q., Xie, H., Xu, J., Wang, Y., and Zhang, Y. Exploring stroke-level modifications for scene text editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 2119–2127, 2023.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Roy, P., Bhattacharya, S., Ghosh, S., and Pal, U. Stefann: scene text editor using font adaptive neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13228–13237, 2020.

Seitzer, M. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, August 2020. Version 0.3.0.

Team, M. L., Ma, H., Tan, H., Huang, J., Wu, J., He, J.-Y., Gao, L., Xiao, S., Wei, X., Ma, X., et al. Longcat-image technical report. *arXiv preprint arXiv:2512.07584*, 2025.

Team, S. I., Qiao, C., Hui, C., Li, C., Wang, C., Song, D., Zhang, J., Li, J., Xiang, Q., Wang, R., et al. Firered-image-edit-1.0 technical report. *arXiv preprint arXiv:2602.13344*, 2026.

Tuo, Y., Xiang, W., He, J.-Y., Geng, Y., and Xie, X. Anytext: Multilingual visual text generation and editing. *arXiv preprint arXiv:2311.03054*, 2023.

Tuo, Y., Geng, Y., and Bo, L. Anytext2: Visual text generation and editing with customizable attributes. *arXiv preprint arXiv:2411.15245*, 2024.

Wang, T., Liu, T., Qu, X., Wu, C., Liu, L., and Hu, X. Glyphmastero: A glyph encoder for high-fidelity scene text editing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 28523–28532, 2025a.

Wang, Y., Zhang, W., Xu, H., and Jin, C. Dreamtext: High fidelity scene text synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 28555–28563, 2025b.

Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.-m., Bai, S., Xu, X., Chen, Y., et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Xie, Y., Zhang, J., Chen, P., Wang, Z., Wang, W., Gao, L., Li, P., Sun, H., Zhang, Q., Qiao, Q., et al. Textflux: An ocr-free dit model for high-fidelity multilingual scene text synthesis. *arXiv preprint arXiv:2505.17778*, 2025.

Yang, Q., Huang, J., and Lin, W. Swaptext: Image based texts transfer in scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14700–14709, 2020.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5505–5514, 2018.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4471–4480, 2019.

Zeng, W., Shu, Y., Li, Z., Yang, D., and Zhou, Y. Textctrl: Diffusion-based scene text editing with prior guidance control. *Advances in Neural Information Processing Systems*, 37:138569–138594, 2024.

Zhang, L., Rao, A., and Agrawala, M. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Zhuang, J., Zeng, Y., Liu, W., Yuan, C., and Chen, K. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. In *European Conference on Computer Vision*, pp. 195–211. Springer, 2024.

Supplementary Material

A. Full Numeric Results

A.1. Comparison with General Image Editing Models

We additionally compare our method with several recent general-purpose image editing models, including Qwen-Image-Edit (Wu et al., 2025), Longcat-Image-Edit (Team et al., 2025), and FireRed-Image-Edit (Team et al., 2026). Since these methods are designed for instruction-based editing rather than mask-guided editing, direct comparison is not entirely straightforward. To enable evaluation, we convert our task into an instruction-based setting using prompts such as: "replace the original text 'XXX' with 'YYY'."

Method	English				Chinese			
	Seq. Acc	NED	FID	LPIPS	Seq. Acc	NED	FID	LPIPS
Qwen-Image-Edit-2509	0.6902	0.7422	24.20	0.1985	0.6288	0.6982	26.02	0.1985
Qwen-Image-Edit-2511	0.7082	0.7893	23.95	0.1874	0.6623	0.7102	24.29	0.1832
Longcat-Image-Edit	0.6725	0.7210	25.60	0.2050	0.6015	0.6750	27.80	0.2070
FireRed-Image-Edit-1.1	0.7375	0.8125	21.80	0.1720	0.6880	0.7355	22.90	0.1705
Ours	**0.8857**	**0.9568**	**7.62**	**0.0365**	**0.8249**	**0.9147**	**7.95**	**0.0268**

Table 4. Comparison with general image editing models on English and Chinese benchmarks.

Table 4 show a consistent ranking across different settings:Ours > FireRed > Qwen-2511 > Qwen-2509 > LongCat.

Our method achieves superior performance in text accuracy, localization, and stylistic consistency. We attribute this advantage mainly to the explicit mask-guided formulation, which directly constrains the editable region and avoids the implicit region inference required by instruction-based methods. In contrast, general editing models often suffer from localization and semantic errors, such as editing incorrect regions, incomplete replacement, unintended synonym substitution, or failure to preserve the original typography and layout. These issues become more evident in multilingual scene text editing, where accurate glyph rendering and precise spatial control are critical.

A.2. Complete Experimental Results on the MSTEdit Dataset

The complete experimental results corresponding to Figure 4 are presented in Table 5. We additionally include OCR-based baselines, including AnyText2 and TextCtrl. However, these methods are not specifically designed for multilingual scene text editing, and their OCR modules mainly support Chinese and English text, resulting in relatively limited performance on broader multilingual editing scenarios.

Language	AnyText2		TextCtrl		FluxText		TextFlux		Ours	
	Seq. ACC	FID	Seq. ACC	FID	Seq. ACC	FID	Seq. ACC	FID	Seq. ACC	FID
Arabic	0.108	75.2	0.110	74.9	0.698	33.5	0.705	30.0	**0.762**	**23.0**
Japanese	0.074	78.8	0.073	79.1	0.604	32.0	0.610	29.0	**0.670**	**22.0**
Korean	0.086	80.2	0.088	79.8	0.602	35.0	0.610	31.5	**0.678**	**24.0**
French	0.227	70.1	0.226	71.4	0.735	32.5	0.744	29.8	**0.808**	**22.0**
German	0.216	71.6	0.218	72.3	0.742	34.5	0.738	30.8	**0.805**	**22.5**
Italian	0.220	70.5	0.219	71.7	0.730	31.0	0.735	28.5	**0.800**	**21.5**
Bengali	0.093	80.7	0.094	80.4	0.708	34.0	0.710	30.2	**0.785**	**23.3**
Hindi	0.100	76.2	0.099	76.4	0.702	30.5	0.713	27.8	**0.790**	**21.0**
Russian	0.087	78.4	0.089	78.1	0.705	34.0	0.708	30.9	**0.775**	**24.5**
Thai	0.061	81.5	0.060	81.8	0.588	32.0	0.595	29.0	**0.672**	**22.0**
Swahili	0.204	75.2	0.206	76.9	0.700	36.0	0.706	32.0	**0.772**	**25.0**
AVG	0.0978	77.9	0.0984	77.9	0.68	33.2	0.69	29.6	**0.76**	**22.8**

Table 5. Multilingual text editing performance comparison across different methods.

The complete experimental results corresponding to Figure 5 are presented in Table 6. Following the predefined language

Table 6. Seq. ACC of initially learned languages under incremental multilingual training. Languages are added sequentially in the following order: *Arabic* → *English* → *French* → *Chinese* → *German* → *Korean* → *Japanese* → *Italian* → *Bengali* → *Hindi* → *Russian* → *Thai* → *Swahili*. Each column denotes the performance after introducing an additional language.

Start	**Number of Languages Seen (Sequential Steps)**										
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
Arabic	0.6812	0.6987	0.7121	0.7094	0.7113	0.7215	0.7198	0.7517	0.7589	0.7612	0.7596
French	0.7423	0.7491	0.7486	0.7615	0.7592	0.7713	0.7689	0.7917	0.7888	0.7909	0.8124
German	0.7096	0.7112	0.7089	0.7123	0.7216	0.7288	0.7321	0.7819	0.7897	0.8014	0.8098
Korean	0.5931	0.6017	0.6024	0.6392	0.6415	0.6387	0.6511	0.6623	0.6589	0.6698	0.6812
Japanese	0.6094	0.6321	0.6313	0.6422	0.6389	0.6415	0.6398	0.6427	0.6411	0.6589	0.6713
Italian	0.7315	0.7398	0.7422	0.7489	0.7716	0.7687	0.7812	0.7794	0.7821	0.7986	0.8015
Bengali	0.7089	0.7115	0.7312	0.7298	0.7421	0.7389	0.7417	0.7719	0.7794	0.7812	0.7786
Hindi	0.6917	0.6994	0.7215	0.7189	0.7486	0.7521	0.7589	0.7697	0.7723	0.7688	0.7912
Russian	0.7113	0.7098	0.7124	0.7391	0.7415	0.7389	0.7422	0.7398	0.7411	0.7617	0.7689
Thai	0.5715	0.5898	0.5912	0.6124	0.6189	0.6215	0.6197	0.6589	0.6612	0.6578	0.6694
Swahili	0.6792	0.6815	0.6789	0.7012	0.6994	0.7117	0.7398	0.7689	0.7713	0.7694	0.7812

list, we progressively introduce new language data in a cyclic order during training.

A.3. Detailed ablation of glyph prompt & style prompt & cooldown

We conduct an ablation study to evaluate different prompt setting, including *text-only*, *text+glyph*, *text+style*, and *text+glyph+style*, across two training stages: pretraining and cooldown. Experiments are performed on both English and Chinese benchmarks. In the cooldown stage, we further compare the effects of paired and unpaired supervision.

The results on Table 7 can be summarized as follows: (1) The text-only setting performs the worst, as no structural or visual reference is provided. (2) In pretraining, text+glyph achieves the best performance, while adding the style prompt degrades results, likely because the source and target texts are identical, which leads to overfitting and copying behavior. (3) In the cooldown stage, text+glyph under both w/o paired and w/ paired yields comparable performance, indicating a continuation of pretraining; text+style provides limited gains due to missing structural guidance, whereas combining glyph and style with paired data yields the most significant improvement.

These findings motivate our strategy of glyph-only prompt on self-supervised data, followed by joint glyph and style prompt with paired data in the cooldown stage.

Language	**Setting**		**Pretrain Stage**		**Cooldown Stage**			
					w/o paired		w/ paired	
	Glyph	Style	Seq. ACC	FID	Seq. ACC	FID	Seq. ACC	FID
English	×	×	0.4127	28.73	-	-	-	-
	✓	×	0.8738	11.56	0.8788(+0.005)	11.52(+0.040)	0.8785(+0.005)	11.55(+0.010)
	×	✓	0.5236	22.48	0.5243(+0.001)	22.48(-0.001)	0.5248(+0.001)	22.48(-0.001)
	✓	✓	0.5412	18.20	0.5440(+0.003)	18.22(-0.019)	0.7982(+0.257)	10.69(+7.511)
Chinese	×	×	0.3563	31.28	-	-	-	-
	✓	×	0.8125	12.01	0.8176(+0.005)	11.97(+0.040)	0.8169(+0.004)	12.01(+0.000)
	×	✓	0.4687	24.16	0.4678(-0.001)	24.19(-0.030)	0.4695(+0.001)	24.12(+0.040)
	✓	✓	0.4879	19.43	0.4911(+0.003)	19.47(-0.040)	0.7426(+0.255)	11.82(+7.610)

Table 7. Ablation study of glyph and style prompts on English and Chinese benchmarks. Glyph and Style denote the glyph and style prompts, respectively. w/o paired indicates training with self-supervised (unpaired) data, while w/ paired denotes training with paired data.

B. Visualization Results

B.1. Out-of-vocabulary (OOV) Instances

Manually design characters. We further manually design a set of synthetic symbol characters that do not exist in real-world writing systems and use them as target texts for evaluation. Figure 8 show that our method still achieves visually coherent and structurally plausible editing results on these previously unseen characters. This observation suggests that our method primarily learns stroke-level compositional representations rather than relying solely on character-level memorization. Consequently, even for non-existent characters, the model can still generate reasonable glyph structures as long as they share common stroke patterns with existing characters.

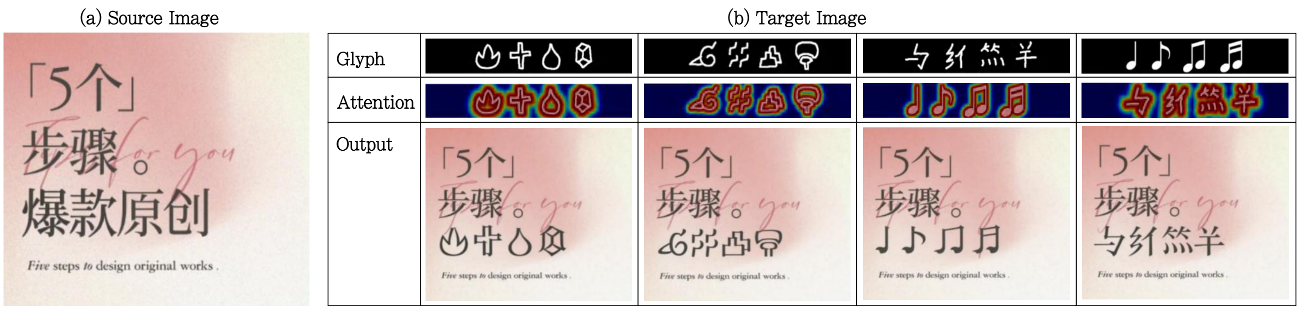

Figure 8. Results of manually designed stroke editing.

Out-of-vocabulary (OOV) characters. We collect 537 rare characters that are not supported by standard OCR vocabularies and conduct a dedicated evaluation on these OOV cases. The complete OOV vocabulary collected in this work is illustrated in Figure 9. Figure 10 results demonstrate that our method still maintains strong performance on the OOV vocabulary, even though these characters rarely appear in existing datasets.

熙 涧 拂 晓 畤 笡 頸 領 哏 炘 旻 貳 陸 横 鄂 別 圪 擋 紙 擁 睥 鬆 蕞 襪 効 燊 圓 頻 頁 繩 韓 萬 顆 霈 俶 週 涼 認 証 垆 專 屬 幾 個 剛

過 與 觀 積 軟 沚 蝦 漬 劑 應 · 錢 垱 輪 閒 : 吳 遺 錘 鍵 轉 變 臉 筆 漸 選 擇 曈 勞 囯 鏤 撥 攜 檯 薵 硌 胼 胝 凊 栊 蒨 釋 嚴 劼 禄

浛 铯 垧 夾 崀 埒 旰 掛 滾 額 脫 舣 茟 揮 瑪 驚 異 琍 蘿 円 隊 團 崑 曌 猊 簾 嶝 畑 惗 歐 鋁 螭 苌 僅 薬 轅 幣 債 緊 濾 擬 掃 嘭 櫻 製

鶉 煩 惱 擼 疊 屳 鉤 闆 績 嗎 慱 毜 赟 鲣 汸 搶 購 徑 鑰 編 焅 餾 側 織 髒 鋆 皃 県 綿 攏 觸 絢 聽 珅 練 責 緩 珮 塝 紮 鏽 奐 撫 盪 倮

贠 杺 琤 槃 議 镪 泇 擊 棄 艷 滅 杪 瀝 誤 镠 钤 焜 麼 茲 玢 戲 償 晉 啷 貞 竑 瀾 烔 規 則 锺 勛 齡 懸 蠶 說 堠 瑧 幫 邊 膩 潡 荛 霧 噴

險 媞 襯 葍 埇 鈞 蕓 訫 櫪 堉 撐 礎 璠 貓 盡 鳴 謙 摺 鉤 鉑 穩 詳 碼 項 擺 圍 繞 瑤 準 糧 黉 遙 塵 荳 熇 剜 侣 婥 戰 覽 饋 兇 蒯 況 –

転 沢 垵 沒 啞 隰 鋰 荳 銅 甑 ³ ² 磙 囊 蔴 檢 锃 – 尋 苿 筌 橦 種 鍾 癢 歆 暝 綠 琻 撲 鬱 絨 腳 搸 牝 尅 懼 顛 嘍 崄 球 渾 杼 傑 癬 鮐

繃 坼 镕 鐙 湑 貯 愷 譚 俥 鄠 泖 吽 馳 埤 . 氿 澴 珖 躍 汘 耊 莀 忞 咏 汆 礙 災 鈦 虢 囧 增 泃 註 洧 鄉 潍 歲 髭 驅 搡 樺 杓 懶 疍 暐

苧 鯿 摜 鈍 箄 詠 壓 螢 劢 璘 須 臟 擔 縛 讓 韪 賞 勵 願 淸 貸 舖 閉 齒 鏈 靚 蘋 溼 恆 酚 烷 胺 氨 韌 舐 沓 缑 焓 塄 鑾 儲 餌 擴 締 遲

遷 鹁 怼 淸 畊 鲥 嶒 俛 閱 薅 氹 稅 呲 袗 潡 洎 衝 賦 宮 隻 籠 仵 掃 蠟 沺 餵 淺 絕 菴 犖 佈 葉 蔣 蟲 蝺 詞 賽 攞 葯 廣 託 遞 疊 鉆 雖

昝 摳 廋 楫 椇 瘡 廢 眾 磻 徂 洹 澔 骈 龜 蕖 痞 蠊 噸 玠 驵 墜 髈 獵 濕 垲 妧 栲 儘 繳 泙 樘 蒺 姺 铁 嫄 塆 埝 垞 騏 溝 鰻 垟 嬉 茈 湻

瓏 栩 穢 嘚 翃 翀 翹 雑 絧 筦 浡 涴 油 巔 餗 锖 瀛 醯 鹼 妠 塱 yyyy 淏 莨 鹨 瀌 獾 侈 — ＿ γ < ↔ λ 丌 Ⅰ ≦ ≧ ∢ ⋗ ©

Figure 9. Out-of-vocabulary character set.

Figure 10. Out-of-vocabulary text editing results. The top-left panel shows the input image, where the original text reads ”清仓价处理”.

B.2. Multilingual Text Editing

For the languages covered in MSTEdit, we provide additional visualization results of our method on several representative language groups, including English/Chinese (Figure 11), Japanese/Korean (Figure 12), and Thai/Russian (Figure 13). The red boxes indicate the target mask regions for text editing.

B.3. Performance on long text

Using Chinese and English as representative examples, we present text editing results with text lengths ranging from 2 to 20 characters. As shown in Figure 14, the performance gradually degrades when the text length exceeds 16 characters, mainly reflected in artifacts such as stroke distortion and reduced structural consistency. For even longer text editing scenarios, segmented editing provides a practical solution. Nevertheless, compared with existing approaches, our method still supports substantially longer text editing while maintaining favorable visual quality and textual consistency.

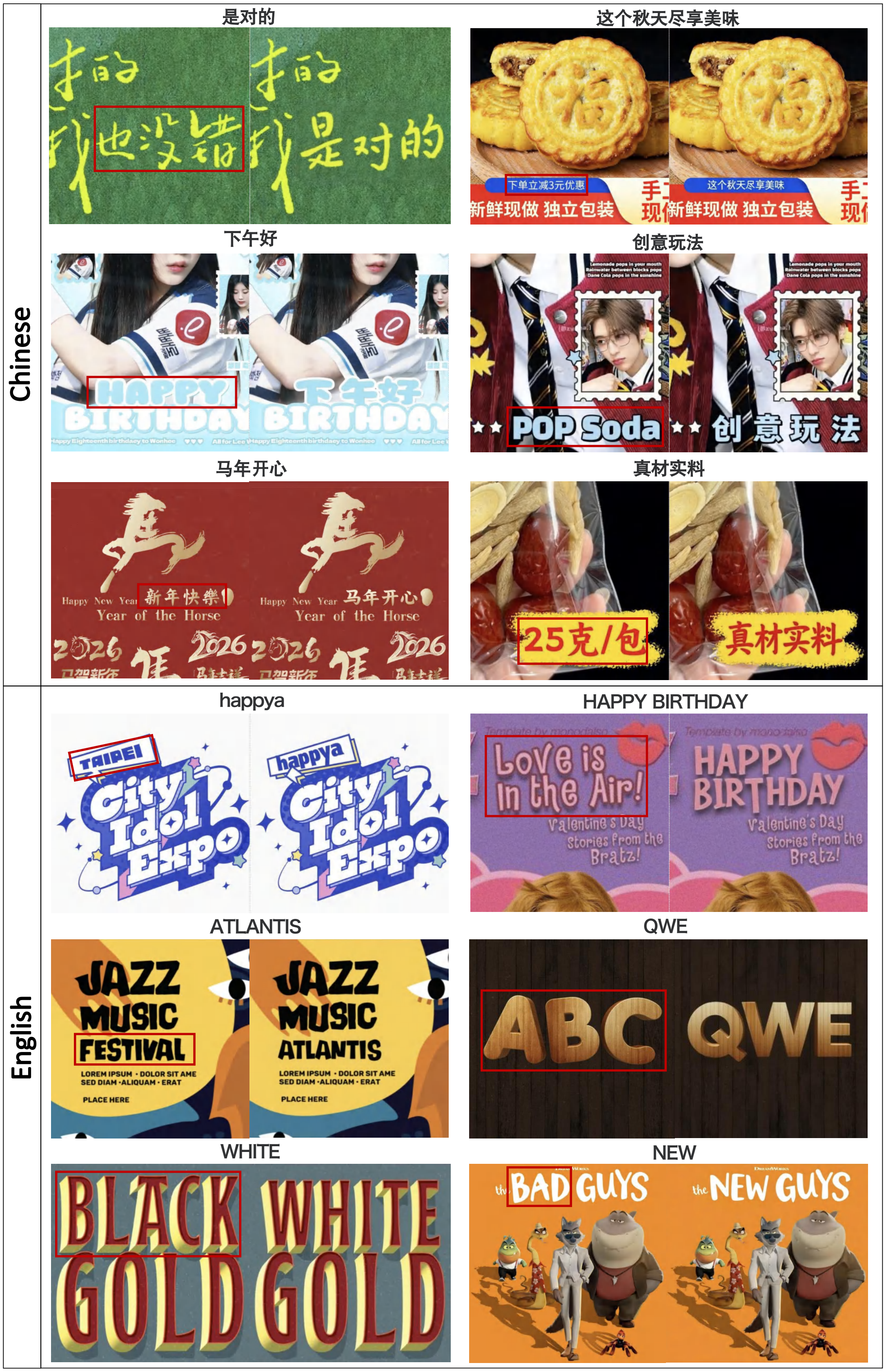

Figure 11. Qualitative results on chinese and english text editing.

Figure 12. Qualitative results on japanese and korean text editing.

Figure 13. Qualitative results on thai and russian text editing.

Figure 14. Text editing results under different text lengths.

C. Paired Dataset Construction for Cooldown Training

The paired dataset used in the cooldown stage is constructed using Nano Banana Pro based on real-world images containing at least one visible text instance. For each image, a single target text region is selected and edited to form an original–edited image pair. In total, more than 15,000 candidate pairs are generated, restricted to Chinese and English text editing scenarios. To ensure high-quality supervision, all generated samples undergo a rigorous human filtering process based on three criteria:

- **Editing correctness:** the source text must be accurately replaced by the target text.
- **Style consistency:** the edited text should preserve the original font, texture, color, and other visual attributes.
- **Content preservation:** all non-target regions should remain unchanged.

After filtering, annotators further annotate the edited text regions with bounding boxes, which are used to derive training masks. The final dataset contains 4,000 high-quality image pairs selected from over 15,000 candidates, with more than 11,000 samples discarded during filtering. The entire process involves 10 annotators over approximately two months.

D. Failure Cases

The failure cases of our method mainly fall into three scenarios: few-to-many editing, many-to-few editing, and multi-line text editing.

- **Few-to-many editing.** When the mask region originally covers only a small amount of text but the target text contains substantially more characters, the generated characters are forced to fit within the limited mask area, leading to artifacts such as character compression and stroke distortion, as illustrated in Figure 15.
- **Many-to-few editing.** When the mask region originally covers a large amount of text but the target text contains significantly fewer characters, excessive blank regions may cause the model to generate hallucinated characters or redundant strokes to adapt to the mask layout, as shown in Figure 16.
- **Multi-line text editing.** When the mask region spans multiple lines of text, the generated results often fail to preserve a proper multi-line layout, as shown in Figure 17. This limitation mainly arises because the glyph prompts are rendered in a single-line format, preventing the model from adaptively performing line wrapping during generation.

Figure 15. Few-to-many character editing results.

Figure 16. Many-to-few character editing results.

Figure 17. Multi-line text editing results.