Recognition-Synergistic Scene Text Editing

Supplementary Material

A. Summary

This supplementary material comprises four components: (1) detailed descriptions of MMPD in our **RS-STE**; (2) implementation details of the fine-tuning stage of detokenizer and data augmentation for recognition; (3) additional ablation studies on model size and the feature representation approach; (4) limitation and analysis; and (5) more visualization examples generated by various scene text editing methods and our **RS-STE**.

B. Details of MMDP

As described in Section 3, the input of MMPD can be denoted as $[\mathbf{E}^{\text{t}}_{T_B}, \mathbf{E}^{\text{i}}_{I_A}, \mathbf{E}^{\text{t}}_{\text{query}}, \mathbf{E}^{\text{i}}_{\text{query}}] \in \mathbb{R}^{2(L+N)\times C}$, where L presents the length of the text embeddings and N presents the length of the flattened image embeddings. In our configuration, we set $L = 32$, $N = 256$ and $C = 384$. Our MMPD consists of 12 transformer blocks, each of which includes a layer normalization layer, causal self-attention with 6 heads, and a fully connected layer.

C. More Implementation Details

Fine-tuning Stage of Detokenizer. We initialize the pre-trained VAE from LDM [5] using configuration parameters $f = 4$, $Z = 8192$ and $d = 3$. To improve the decoder's performance in reconstructing text images from continuous features, we fine-tune the VAE on our training datasets. Specifically, we remove the codebook-related components from the pre-trained model and train it for 100k iterations using the Adam [3] optimizer with a batch size of 256, and a learning rate of 1.25×10^{-3}. The reconstruction performance of the VAE before and after fine-tuning on the evaluation dataset Tamper-Syn2k and ScenePair is shown in Tables 6 and 7. Compared to the pre-trained VAE, the fine-tuned VAE demonstrates better image reconstruction performance for text images. This metric also indicates the upper limit of the image editing performance when using the VAE decoder as an Image Detokenizer.

Details of Data Augmentation for Recognition. To evaluate the effectiveness of our data augmentation strategy, we use the Union14M-L dataset on classical recognition model ABINet [1], and state-of-the-art recognition model MAERec-S [2]. We compare our method with MOSTEL [4] to validate its superiority. For instance, on the ABINet [1] model, we first evaluate the pre-trained ABINet [1] on the Union14M-L dataset by testing on its evaluation set and identifying cases of incorrect recognition ("bad cases"). These bad cases are then modified using our method or MOSTEL [4], generating additional text images that maintain a similar style but contain varied content for further fine-tuning of ABINet [1]. We visualize some of the targeted augmented data generated by **RS-STE** in Figure 6.

In implementation, we employ each scene text editing model to randomly generate five variations per bad case, creating images with different textual content while retaining the original style. Subsequently, we utilize the corresponding pre-trained recognition models to recognize the generated targeted augmented images. Any data with an edit distance between the recognition result and the ground truth exceeding one-third of the word length is discarded. This process results in about 250k and 170k augmented images for ABINet [1] and MAERec-S [2], respectively. The models are subsequently fine-tuned on a combination of these augmented datasets and the Union14M-L dataset.

Table 6. The image reconstruction performance of VAE before and after fine-tuning on Tamper-Syn2k.

Fine-tune	Tamper-Syn2k			
	MSE↓	PSNR↑	SSIM↑	FID↓
✗	0.00453	25.22	83.17	30.91
✓	**0.00049**	**34.01**	**98.57**	**13.34**

Table 7. The image reconstruction performance of VAE before and after fine-tuning on ScenePair.

Fine-tune	ScenePair			
	MSE↓	PSNR↑	SSIM↑	FID↓
✗	0.00169	29.77	90.87	19.34
✓	**0.00064**	**34.00**	**97.26**	**4.10**

D. More Ablation Studies

D.1. Effect of Model Size on Performance

To further investigate the effect of model size on editing performance, we conduct experiments using an 85.5M MMDP model, configured with an embedding dimension of 768 and 12 attention heads. The results, presented in Table 8, demonstrate that increasing the model size significantly enhances the text editing capabilities of our approach. Therefore, in practical application, different model configurations can be selected based on a trade-off between computational resources and performance requirements.

Table 8. The impact of model scaling on editing performance of **RS-STE**. 'Tiny' denotes the 22.5M MMDP while 'Small' denotes the 85.5M one.

Model	MMDP	Tamper-Syn2k				Tamper-Scene	ScenePair				
	#Param.	MSE↓	PSNR↑	SSIM↑	FID↓	RecAcc↑	MSE↓	PSNR↑	SSIM↑	FID↓	RecAcc ↑
RS-STE-Tiny	22.5M	0.0076	22.54	72.90	**30.29**	86.12	0.0267	17.35	46.09	41.37	**91.80**
RS-STE-Small	85.5M	**0.0072**	**22.87**	**73.18**	31.34	**94.14**	**0.0254**	**17.55**	**46.97**	**39.13**	91.56

Table 9. The image reconstruction performance of continuous VAE and discrete VAE.

Condition	Tamper-Syn2k				ScenePair			
	MSE↓	PSNR↑	SSIM↑	FID↓	MSE↓	PSNR↑	SSIM↑	FID↓
discrete	0.00146	30.18	93.79	21.35	0.00080	32.88	96.74	4.55
continuous	**0.00049**	**34.01**	**98.57**	**13.34**	**0.00064**	**34.00**	**97.26**	**4.10**

Figure 6. The visualization of targeted augmented data generated by **RS-STE** from bad cases of recognition model ABINet [1].

Figure 7. Editing results of different methods on curved text.

Table 10. Text image editing performance with discrete and continuous feature representation methods.

Methods	Tamper-Syn2k			
	MSE↓	PSNR↑	SSIM↑	FID↓
discrete	0.0167	19.03	70.57	46.73
continuous	**0.0076**	**22.54**	**72.90**	**30.29**

D.2. Discrete Feature Representation

Since the pre-trained VAE from LDM [5] utilizes Vector Quantization [6], we also retain the fine-tuned VQ-VAE in our approach, using its encoder and codebook as the tokenizer and its decoder as the detokenizer. This design enables training on the discrete representations of both the source image and target text, leveraging the VAE's encoding and decoding mechanisms to their full potential. However, as illustrated in Table 10, our results indicate that the

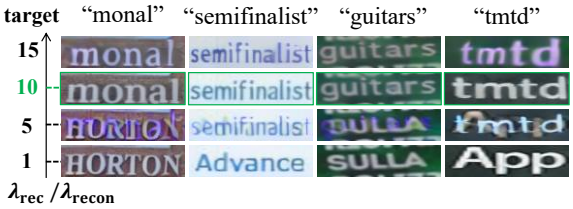


Figure 8. Visualization examples of different ratio of recognition loss weight λ_{rec} and reconstruction loss weight λ_{recon}.

discrete feature encoding approach performs worse than the continuous encoding strategy adopted in our method.

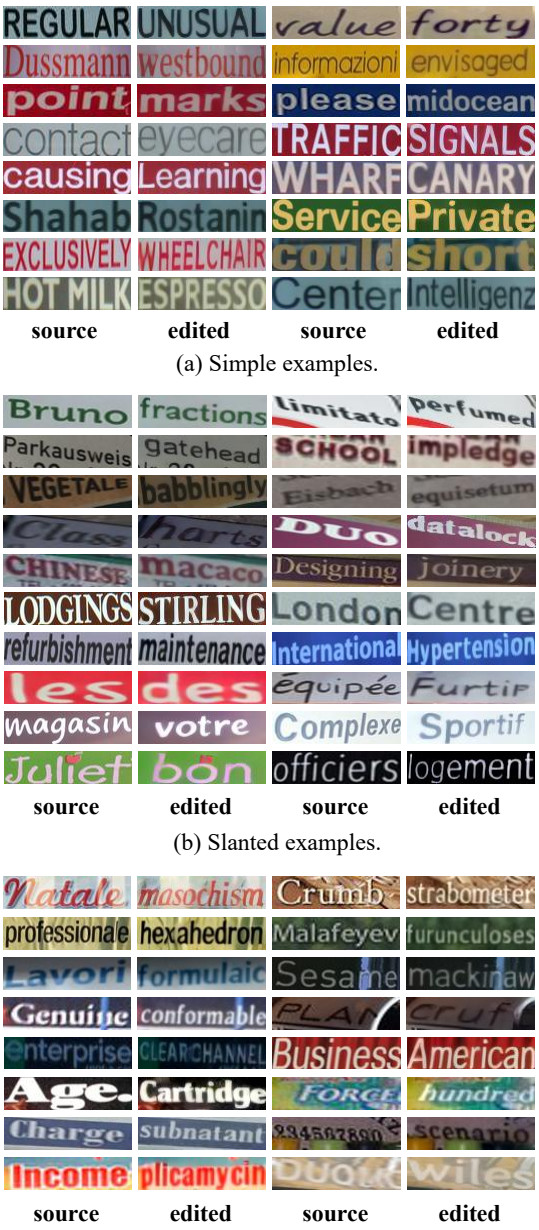
This can primarily be attributed to two factors: (1) The discretization of images introduces information distortion, resulting in poorer reconstruction quality compared to continuous representations. As shown in Table 9, for the given dataset, the reconstruction performance of the discrete form is inferior to that of the continuous form. (2) Continuous representations effectively mitigate the inherent decoding bias of the detokenizer. As discussed in Section 3, for continuous image features, reconstruction loss can be computed on the detokenized images, ensuring pixel-level accuracy in the final output. In contrast, for discrete representations, supervision can only be applied to the discretized image features decoded by the MMPD, leading to feature distortions during the detokenization process.

D.3. Loss Weights

In the cyclic training stage described in Section 3.3, we observe that the ratio of the recognition loss weight, defined as $\lambda_{\text{rec}} = (\lambda_6 + \lambda_7)/2$, to the image reconstruction loss weight, defined as $\lambda_{\text{recon}} = (\lambda_4 + \lambda_5)/2$, plays a crucial role in ensuring content and style consistency. Consequently, we conduct an ablation study to examine the effects of varying this ratio, as shown in Figure 8. Our findings indicate that a ratio close to 10 consistently produces high-quality images.

E. Limitation and Analysis

A potential limitation of our method as well as most other methods for scene text editing lies in the limited performance when editing images with extremely large text curvature, as shown in Figure 7. This limitation is mainly attributed to the scarcity of such data in synthetic training data. To further investigate this issue, we train our model with additionally synthetic curved text samples generated using the synthesis engine mentioned in Section 4.1, and our method (**RS-STE+**) achieves robust curved text editing, which implies that such limitation arises from insufficient training data of curved text.



(a) Simple examples.

(b) Slanted examples.

(c) Examples with complex backgrounds.

Figure 9. More visualization examples edited by **RS-STE** on unpaired real-world dataset Tamper-Scene.

F. Visualization Examples of RS-STE

To further demonstrate the superiority of our **RS-STE**, we include additional visualization results of the text images before and after editing with **RS-STE**, as illustrated in Figure 9.

References

[1] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In *CVPR*, pages 7098–7107, 2021. 1, 2

[2] Qing-Yuan Jiang, Jiapeng Wang, Dezhi Peng, Chongyu Liu, and Lianwen Jin. Revisiting scene text recognition: A data perspective. *ICCV*, pages 20486–20497, 2023. 1

[3] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[4] Yadong Qu, Qingfeng Tan, Hongtao Xie, Jianjun Xu, Yuxin Wang, and Yongdong Zhang. Exploring stroke-level modifications for scene text editing. In *AAAI*, pages 2119–2127, 2023. 1

[5] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1, 2

[6] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *NIPS*, 30, 2017. 2