UDiffText: A Unified Framework for High-quality Text Synthesis in Arbitrary Images via Character-aware Diffusion Models

Yiming Zhao and Zhouhui Lian*

Wangxuan Institute of Computer Technology, Peking University, China
{zhaoym, lianzhouhui}@pku.edu.cn

A More Implementation Details

We have observed that the area proportion of the masked region in a given image significantly impacts text rendering performance. As a result, we enforce strict constraints on the proportions of the text mask and character segmentation mask in our training datasets. Specifically, we filter out images with a text mask proportion less than 1% or a character segmentation mask proportion less than 0.1%. Additionally, we perform image cropping and resizing to ensure uniform input scales and maintain reasonable text region proportions.

For images in LAION-OCR, character-level segmentation maps are derived using the segmentation model proposed in [2]. This model is not instance-based and thus can not distinguish different instance regions of a specific character. Besides, the segmentation model may produce unsatisfactory or incorrect results, such as omitting certain characters or partially masking them. To perform data cleaning and augmentation, we initially employ connected components extraction to separate repeated characters in the binary masks, thereby providing precise positional information and eliminating ambiguity in attention map constraints. Subsequently, we apply a morphological opening operation to eliminate noise points and use morphological dilation to slightly expand the masked character areas to avoid extremely small scale. An illustrative example of the data augmentation process can be seen in Fig. 1. It should be noted that the issue of missing characters in segmentation maps cannot be completely resolved and may adversely affect the text rendering performance of our model.

B More Comparison Results

We carry out additional qualitative experiments on the scene text inpainting (substitute text) task and compare our results with those of the aforementioned baselines. Further results can be viewed in Fig. 2. It is evident that our method produces the most visually appealing outcomes, distinguished by high text rendering precision and consistency with the visual context.

We also conduct user study for additional validation. To be specific, we randomly choose 15 comparison cases, each encompassing outputs from DiffSTE [3],

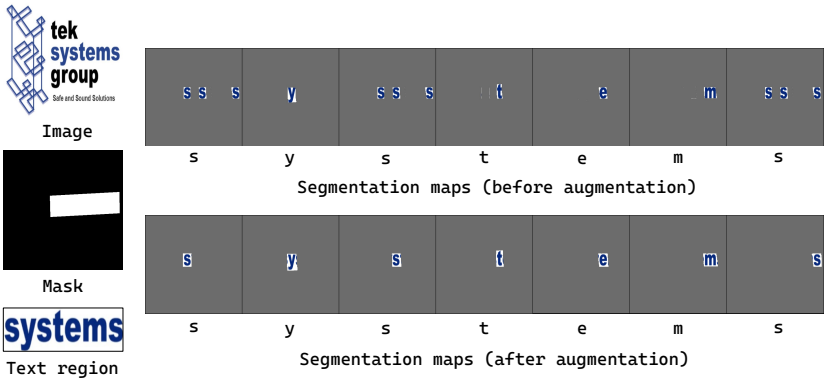

Fig. 1: Data augmentation. The image and the binary mask of the text region can be seen at the left side while the extracted segmentation map for each character is shown at the right side, both before (the first row) and after augmentation (the second row).

Textdiffuser [2] and our model. 20 human evaluators are asked to rank the images based on the metrics of text accuracy and visual consistency. As shown in Tab. 1, we calculate the average win rate based on the feedback and the results again prove the superiority of our method.

Table 1: User Study. We showcase the average win rate of the three methods on both text accuracy and visual consistency.

Metric/Method	DiffSTE	TextDiffuser	Ours
Text Accuracy (%) ↑	23.0	14.0	**63.0**
Visual Consistency (%) ↑	11.8	10.4	**77.8**

C More Application Results

We present additional qualitative results on the previously discussed scene text inpainting (substitute text) task (Fig. 3) and the accurate T2I generation task (Fig. 4). Leveraging the inpainting-based architecture, UDiffText is proficient in generating coherent text in both real-world images and AI-generated images. Consequently, it can serve as an artistic text designer in a variety of graphic design tasks, including poster design and advertisement design.

In the accurate T2I generation task, we utilize off-the-shelf LLM (GPT-3.5) to generate the prompts for first-stage image generation, the prompt we use for each case are as follows:

1. A poster for a movie premiere with the title “Complicated Matrix” and the tagline “The ultimate choice is yours”. The poster has an image of a man

in a black suit and sunglasses holding a gun. The text is in a futuristic and metallic font.
2. A flyer for a yoga class with the title "My Peaceful Zone" and the slogan "Find your balance". The flyer has a white background with green leaves and flowers. The text is in a simple and elegant font.
3. A logo for a coffee shop called "My Favourite cup of coffee". The logo is a stylized coffee bean with a smiley face and sunglasses. The text is in a handwritten and casual font.
4. A book cover for a sci-fi novel called "The Final Frontier". The book cover has an image of a spaceship flying over an alien planet. The text is in a futuristic and metallic font.
5. Create an artistic composition for a nature conservation campaign. Include lush landscapes, endangered species, and the phrase "Preserve Our Planet" in elegant typography.
6. Craft a captivating banner for a technology summit. Use sleek lines, futuristic elements, and include the phrase "Innovate for Tomorrow" in a dynamic font.
7. A poster for a music festival with the title "Fascinating rock and roll stars" and the logo of a guitar. The poster has a colorful background with geometric shapes and patterns. The text is in a bold and funky font.

As shown in Fig. 5, we conduct text editing on more challenging and creative cases generated with Ideogram and the results prove the generalization performance of our method, which can be applied to stylized design text synthesis tasks.

D Dicussion on the Character-level Text Encoder

Our method is inspired by the concept of letter-wise encoding from [4], but there are clear differences. [4] develops their method for T2I generation with legible text, while our method is designed to solve the task of scene text inpainting. In terms of the choice of text encoders, [4] utilizes the heavy-weight ByT5 encoder which requires much GPU memory and computation cost, while we instead leverage a light-weight transformer-based encoder which is trained using contrastive learning. The key point is, our model structure and the design of training objectives help the light-weight text encoder to gain similar ability of high accuracy text rendering compared to a much larger text encoder. This design choice significantly eliminates redundancy and improves efficiency.

Our text encoder is trained using the clip loss together with the cross-entropy loss to get highly distinguishable embeddings to avoid character confusion in model outputs. To verify this insight, we visualize the learned embeddings of each printable character using t-SNE, as shown in Fig. 6. We then measure the pairwise cosine similarity of all the embeddings and calculate the average value. The embeddings learned with the additional cross-entropy loss have a lower average cosine similarity, which make them more separateable and distinguishable.

E Discussion on the Local Attention Control

Here we analyze the superiority of our local attention control against the straightforward segmentation map guidance from two aspects: (1) Task modeling: Methods using segmentation maps (TextDiffuser [2]) or rendered glyph images (GlyphControl [6]) as input treat the text image synthesis task as an I2I-like transfer procedure, where the model learn to create a mapping from the direct content condition to the rendered text, which can introduce lack of style variety (text font and texture). In contrast, we treat it as a conditional generation procedure and train our model to learn the layout distribution and appearance of each character (stored in the Keys and Values of the CA module). The segmentation maps supervise the model to focus on the limited region of each character to learn a more precise distribution. Quantitative results have proven that our approach achieves better text rendering precision and image quality compared to others. (2) Model interpretability: It is evidenced that the CA maps are essential in determining the spatial layout of objects in generated images. With the help of $\mathcal{L}_{loc}$, we can interpret the CA maps of our trained model as the precise ROI of rendered characters, which reflect the existence and position of each character and provide a more flexible control manner for text rendering.

To provide a more intuitive demonstration of the proposed local attention constraint, we present additional visualization results in Fig. 7 and Fig. 8. For a specific text rendering case, we extract the attention maps from the middle block of the U-Net at an intermediate sampling step. It is clear that under the constraint of our local attention loss, the model focuses on the specific region of each character. The attention values are high and centralized in the character areas, while they are nearly zero in areas of no concern. This type of constraint assists the model in concentrating on learning the visual features of characters rather than irrelevant textures. Furthermore, we up-sample the attention maps to the scale of the output image and obtain segmentation maps of each generated character, as shown in the last column. This experiment illustrates a potential application of text segmentation based on our trained model and corresponding image editing methods with diffusion models.

F Discussion on the Length of the Synthesised Text

According to previous work [5], English words with less than 13 characters cover a total word frequency of 99.025%. Within the LAION-OCR test set, only 1.41% of the annotated words exceed 12 characters in length. While results for longer text may fall short of those reported in the paper, this is of little consequence as our method proves effective for the majority of text inpainting scenarios. In tasks involving long text synthesis, UDiffText can be applied iteratively to generate a text paragraph word by word and line by line. In future work, we can expect a more efficient improvement to generate one text paragraph at one time with the help of additional guidance like line-level layout condition.

G Failure Cases and Limitations

Despite its ability to render coherent text in arbitrary given images, our method can still produce unsatisfactory results, including rendering text with distorted characters, repeated characters, incorrect characters and missing characters, as shown in Fig. 9. These failure cases occur more frequently when the text to be rendered is relatively long or when the masked region is excessively oblique.

Since our model relies on visual context to render the expected text, it may struggle to generate coherent text when the image background is relatively simple. Furthermore, the current version of our method can only satisfactorily handle text sequences with a limited number of characters (up to 12 characters in our implementation). Despite proving effective for the majority of text inpainting scenarios, our method does exhibit limitations in tasks involving long text synthesis, such as paragraph generation or extensive document synthesis. Moreover, due to the lack of a style encoder, our model is unable to generate text with the style well-aligned with the original text in some images, thus cannot be directly applied to tasks like scene text editing. Nonetheless, the proposed method is still enlightening for the scene text editing task and we will leave it as future work to cover the text generation and editing problem with an improved design.

References

1. Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., Manassra, W., Dhariwal, P., Chu, C., Jiao, Y.: Improving image generation with better captions (2023), `https://cdn.openai.com/papers/dall-e-3.pdf`
2. Chen, J., Huang, Y., Lv, T., Cui, L., Chen, Q., Wei, F.: Textdiffuser: Diffusion models as text painters. Advances in Neural Information Processing Systems **36** (2024)
3. Ji, J., Zhang, G., Wang, Z., Hou, B., Zhang, Z., Price, B., Chang, S.: Improving diffusion models for scene text editing with dual encoders. arXiv preprint arXiv:2304.05568 (2023)
4. Liu, R., Garrette, D., Saharia, C., Chan, W., Roberts, A., Narang, S., Blok, I., Mical, R., Norouzi, M., Constant, N.: Character-aware models improve visual text rendering. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Jul 2023)
5. van de Weijer, J., Eeg-Olofsson, M., Sigurd, B.: Word length, sentence length and frequency: Zipf's law revisited. Studia Linguistica **58**(1), 37–52 (2004)
6. Yang, Y., Gui, D., Yuan, Y., Liang, W., Ding, H., Hu, H., Chen, K.: Glyphcontrol: Glyph conditional control for visual text generation. Advances in Neural Information Processing Systems **36** (2024)

Fig. 2: Additional comparison results on the scene text inpainting (substitute text) task. The first row consists of the original images, while the second row comprises the input images with binary masks applied to the text region. The specific word to be generated is indicated at the top of each column.

Fig. 3: Additional application results for scene text inpainting (substitute text) task. The word to be rendered is annotated at the bottom of each case.

Fig. 4: Additional application results for accurate T2I generation task. The first column demonstrates the initial outputs of DALL-E-3 [1] conditioned by the given prompts while the last column shows our final outputs after correcting the text in masked regions. The word to be corrected is annotated at the left of each row.

Fig. 5: Text editing results on more challenging and creative cases. The original images and edited images are shown in the first row and the second row, separately. The edited text of each case is annotated at the bottom of each column.

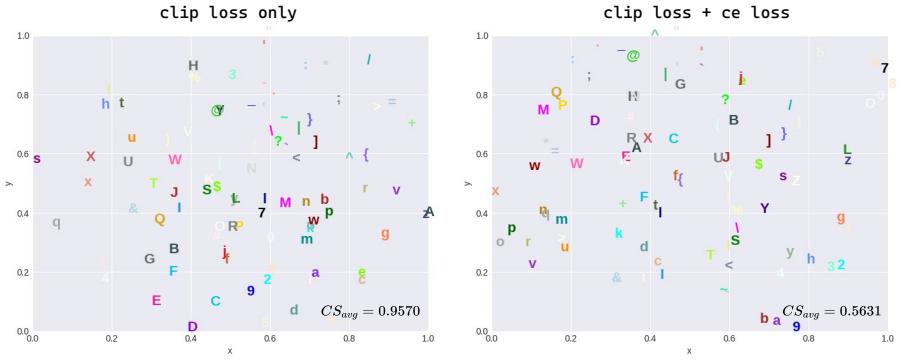

Fig. 6: The t-SNE visualization of the learned character embeddings. The average cosine similarity of the embeddings is annotated at the bottom-right corner.

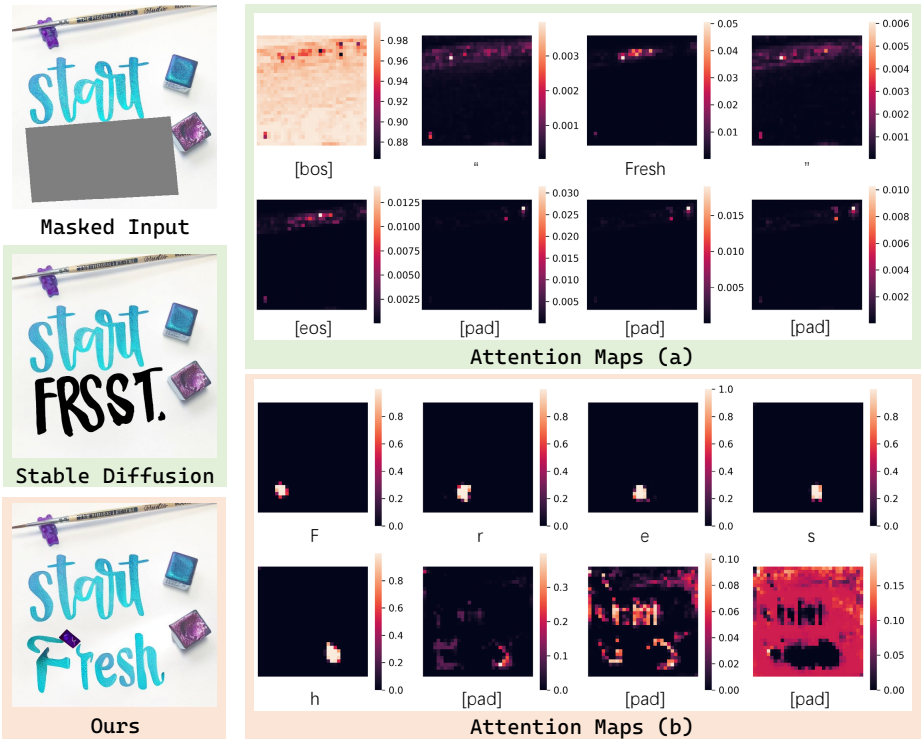

Fig. 7: Additional visualization results. The expected text is "Fresh" and the masked input is displayed at the top left. The attention maps extracted from the U-Net of Stable Diffusion (a) and ours (b) can be observed on the right side. The specific token of each attention map is annotated at the bottom.

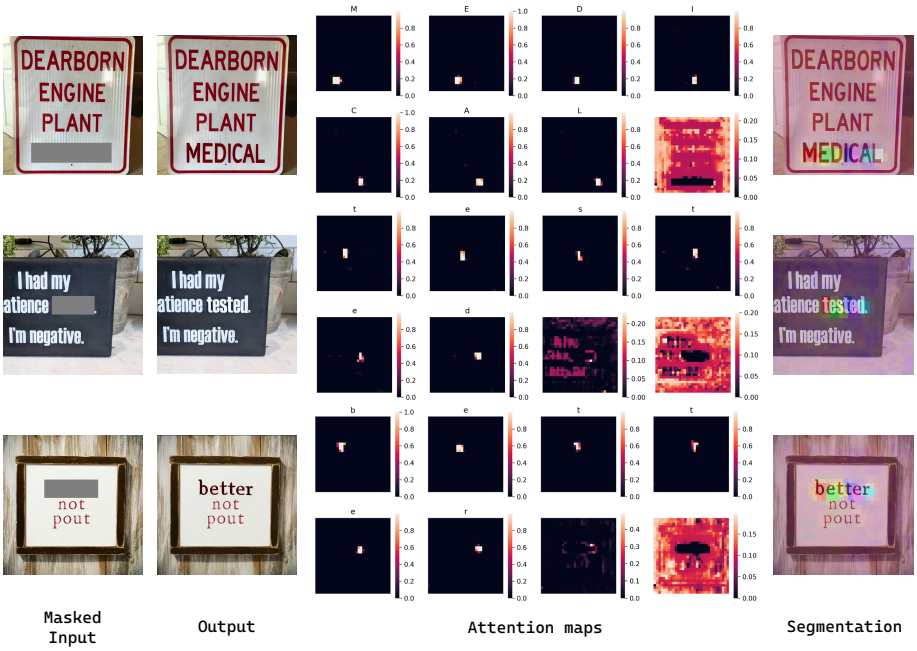

Fig. 8: Additional visualization results. The first column is the masked inputs for our UDiffText while the second column shows the outputs. The attention map of each case is extracted from the middle block of the U-Net at intermediate sampling step. The specific token of each attention map is annotated at the top of each map. We upsample the attention maps to get segmentation maps of the generated images, which are demonstrated at the last column.

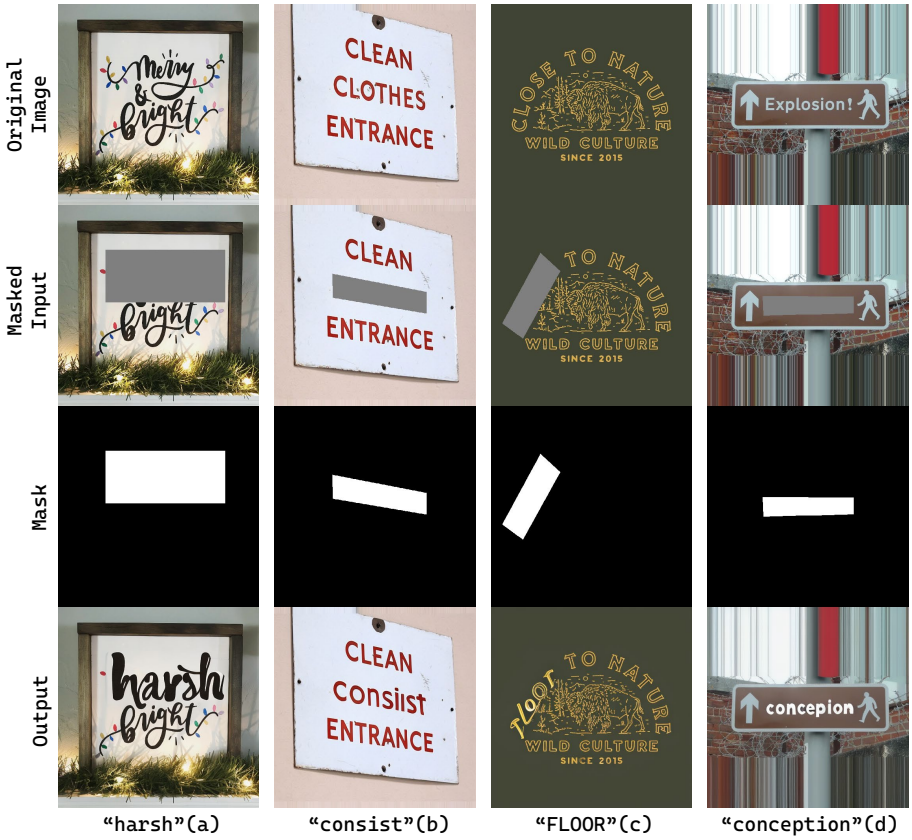

Fig. 9: Failure cases. We show some unsatisfactory results of our method, including distorted characters (a), repeated characters (b), wrong characters (c) and missing characters (d). The word to be rendered is annotated at the bottom of each column.