# DREAMTEXT: High Fidelity Scene Text Synthesis

**Yibin Wang**[1,2], **Weizhong Zhang**[1,4], **Honghui Xu**[5], **Cheng Jin**[1,3†]
[1]Fudan University [2]Shanghai Innovation Institute
[3]Innovation Center of Calligraphy and Painting Creation Technology, MCT
[4]Shanghai Key Laboratory of Intelligent Information Processing [5]Zhejiang University of Technology
yibinwang1121@163.com, weizhongzhang@fudan.edu.cn, xhh@zjut.edu.cn, jc@fudan.edu.cn


Figure 1. Displayed are the results generated using our DREAMTEXT, showcasing its prowess across varied inputs.

## Abstract

*Scene text synthesis involves rendering specified texts onto arbitrary images. Current methods typically formulate this task in an end-to-end manner but lack effective character-level guidance during training. Besides, their text encoders, pre-trained on a single font type, struggle to adapt to the diverse font styles encountered in practical applications. Consequently, these methods suffer from character distortion, repetition, and absence, particularly in polystylistic scenarios. To this end, this paper proposes DREAMTEXT for high-fidelity scene text synthesis. Our key idea is to reconstruct the diffusion training process, introducing more refined guidance tailored to this task, to expose and rectify the model's attention at the character level and strengthen its learning of text regions. This transformation poses a hybrid optimization challenge, involving both discrete and continuous variables. To effectively tackle this challenge, we employ a heuristic alternate optimization strategy. Meanwhile, we jointly train the text encoder and generator to comprehensively learn and utilize the diverse font present in the training dataset. This joint training is seamlessly integrated into the alternate optimization process, fostering a synergistic relationship between learning character embedding and re-estimating character attention. Specifically, in each step, we first encode potential character-generated position information from cross-attention maps into latent character masks. These masks are then utilized to update the representation of specific characters in the current step, which, in turn, enables the generator to correct the character's attention in the subsequent steps. Both qualitative and quantitative results demonstrate the superiority of our method to the state of the art. Our project page is here.*

†Corresponding author.

## 1. Introduction

Scene text synthesis involves modifying or inserting specified text in arbitrary images while maintaining its natural and realistic appearance. Earlier GAN-based style transfer methods [13, 14, 16, 27] accomplish this by transferring the text style from a reference image to the rendered target text image. However, they are constrained in their capacity to generate text in arbitrary styles, such as font and color. These


Figure 2. Current methods encounter significant challenges, i.e., character repetition and absence (top) and character distortion (bottom).

limitations are effectively addressed by utilizing the diffusion model [5, 8, 11, 12, 17, 21, 22], harnessing its inherent prior knowledge and robust capabilities. Nevertheless, these methods still encounter challenges in generating accurate text within polystylistic images due to the inadequate conditional guidance provided by their character-unaware text encoder. To this end, a recent study [29] proposes a character-level text encoder, providing more robust conditional guidance for the diffusion model.

Despite their effectiveness, these methods are still difficult to accurately render text within complex scenes. Specifically, these methods such as UDiffText [29] and TextDiffuser [5] solely utilize a single font type to pre-train the text encoder, which is subsequently employed to fine-tune the generator. However, the diverse font styles encountered in practical applications present a significant challenge: the restricted representation domain hampers their ability to render text with unseen font styles, *e.g.,* cases shown in Fig. 2 (bottom). Furthermore, these methods rely on character segmentation masks to supervise characters' attention for position control. We argue this approach has significant limitations: (1) unlike classification tasks with uniquely determined labels, the optimal generation positions for characters can be diverse. Therefore, using the specific masks to rigidly constrain the model may limit its flexibility in estimating optimal positions, hindering its ability to adapt to varied and complex scenarios. Besides, (2) these masks generated by a pre-trained segmentation model [5] are imprecise, tending to over-segment character regions. As a result, the model encounters issues of character repetition and absence as illustrated in Fig. 2 (top left) and (top right) respectively. We suspect these issues stem from characters' attention maps, as they reflect the underlying response to the character's final generated position [23, 24]. Therefore, we visualize the attention maps of their problematic results by rendering attention directly onto the images. We observe certain characters' attention misalignment, resulting in character duplication (see Fig. 3 (a) and (c)). Besides, excessive dispersion of attention also leads to character absence (see Fig. 3 (b)). This observation underscores the insufficiency of these methods in guiding the model autonomously estimating optimal character-generated position. Further, we also assess specific methods [5, 29] quantitatively on two datasets to evaluate their proficiency in directing characters' attention toward their optimal generation regions throughout multiple global training steps. Specifically, for each test sample, we extract potential position masks for all characters based on their cross-attention maps from the inference process. Then, we compute the mean Intersection over Union (mIoU) between these masks and ground truth character segmentation masks. Their inoptimal results depicted in Fig. 4 further corroborate our hypothesis.

In summary, current methods encounter challenges in accurately rendering text within complex scenes due to their constrained representation domain and synthesizing characters in their ideal regions due to the lack of effective guidance for the model to estimate optimal character-generated position. To this end, this paper proposes DREAMTEXT for high-fidelity text synthesis. We reconstruct the diffusion training process, embedding more refined guidance: we expose and rectify the characters' attention to focus more precisely on their ideal generation areas, and enhance the learning of text regions and character representation by additional constraints. However, this reconstruction inevitably introduces a complex hybrid optimization problem encompassing both discrete and continuous variables. Therefore,

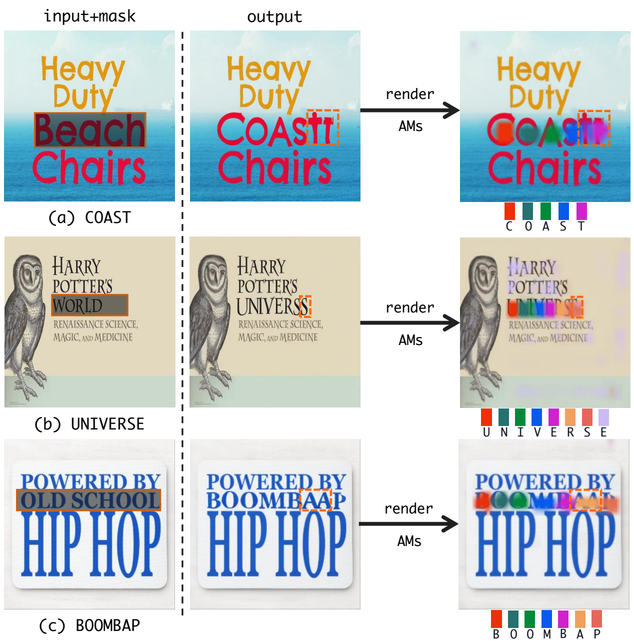


Figure 3. The problematic results rendered by characters' attention maps (AMs).

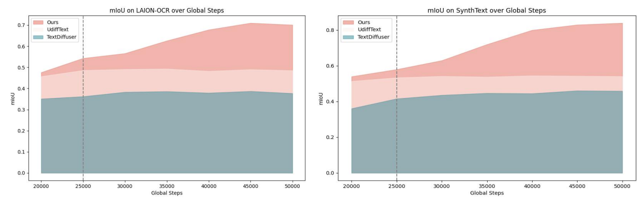


Figure 4. The mIoU scores of UDiffText, TextDiffuser, and our method on LAION-OCR and SynthText over global training steps. Our method adopts a balanced supervision strategy: we initially use latent character masks to steer the character's attention for a warm-up in the earlier stage and stop guiding after 25,000 steps.

we design a heuristic alternate optimization strategy. Concurrently, we jointly train the text encoder and the generator, leveraging the diverse font styles within the training dataset to enrich character representation space. We demonstrate that this joint training can be seamlessly integrated into the heuristic alternate optimization process, facilitating a synergistic interplay between character representations learning and character attention re-estimation. Specifically, in each iteration, the pipeline begins by encoding the potential character-generated position information from cross-attention maps into latent character masks. These masks are subsequently employed to refine the representation of specific characters within the text encoder, facilitating the calibration of the characters' attention in the subsequent step. This iterative process contributes to both learning of better character representation and, as a by-product, explicit guidance for autonomously estimating character position.

Notably, the generator may initially struggle to direct attention to the desired generation region for each character, resulting in suboptimal latent masks and impacting training progress. Unlike existing methods [5, 29] that rigidly constrain character attention, we employ a balanced supervision strategy. Initially, we assist in calibrating the attention using character segmentation masks for a warm-up in the earlier stage. Once the model has preliminarily acquired the ability to estimate ideal generation positions, we remove this guidance, allowing the model to autonomous learning iteratively. This strategy strikes a balance between constraining the model and unleashing its flexibility in estimating optimal generation positions. Fig. 4 illustrates the efficacy of our method in continuously improving the generator's ability to estimate the optimal characters' position.

Our contributions are: (1) The proposed DreamText effectively alleviates the issues of character repetition, absence, and distortion encountered by existing methods. (2) Our heuristic alternate optimization strategy, integrating the joint learning of the text encoder and U-Net, orchestrates a symbiotic relationship between learning character representations and re-estimating character attention. (3) Our balanced supervision strategy strikes a balance between constraining the model and unleashing its flexibility in estimating optimal generation positions. (4) Both qualitative and quantitative results demonstrate the superiority of our method.

## 2. Related Work

Earlier GAN-based methods achieve scene text synthesis by transferring the text style in a reference image to the rendered target text image [10, 16, 18, 20, 27]. Specifically, STEFANN [16] employs a FANnet to edit individual characters and incorporates a placement algorithm to generate the desired word. SRNet [25] and MOSTEL [14], on the other hand, split the task into two main parts: background inpainting and text style transfer. This approach allows for the end-to-end synthesizing of entire words. Despite their simplicity and effectiveness, they are limited in their ability to generate text in arbitrary styles and locations, often producing less natural-looking images.

To this end, several diffusion-based approaches have emerged to address these challenges. These approaches harness the robust capabilities of diffusion models to edit or generate scene text, thereby enhancing the quality and diversity of the generated content. DiffSTE [8] proposes a dual encoder structure, comprising a character text encoder and an instruction text encoder. These components are employed for instruction tuning, providing improved control over the backbone network. DiffUTE [4] employs an OCR-based glyph encoder to extract glyph guidance from the rendered glyph image. TextDiffuser [5] concatenates the segmentation mask as conditional input and employs character-aware loss to control the generated characters precisely. However, they still face challenges in generating accurate text

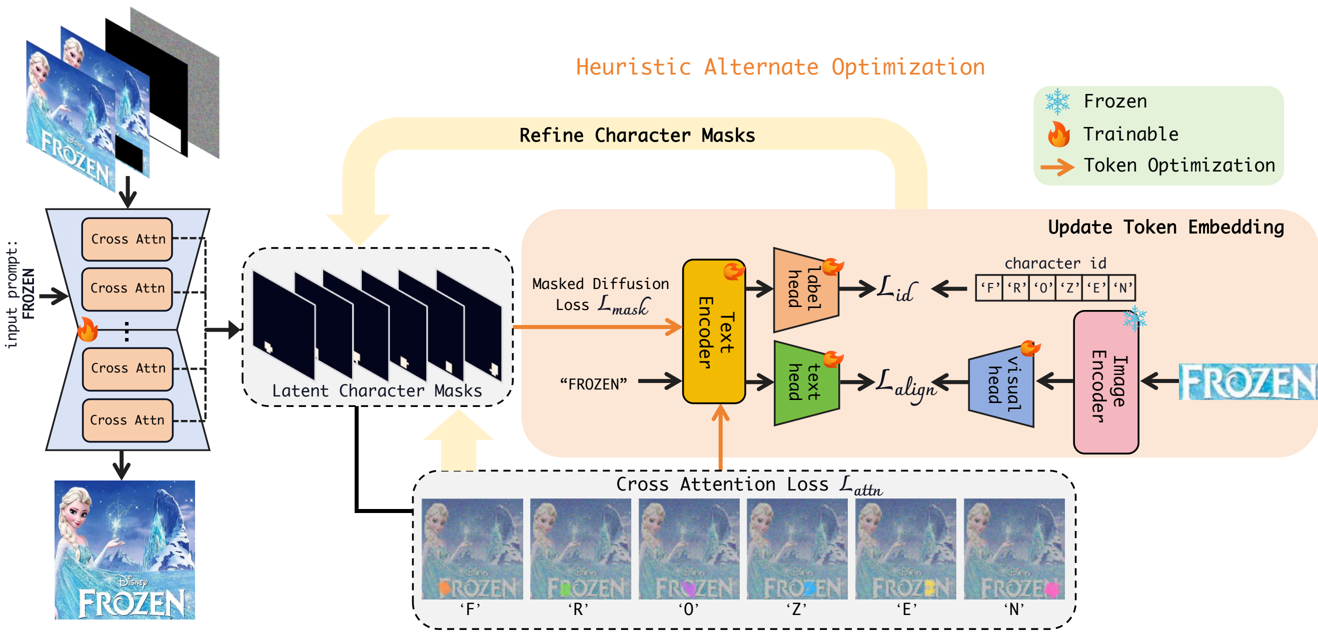


Figure 5. An overview of proposed heuristic alternate optimization strategy.

within polystylistic images, primarily due to the insufficient conditional guidance provided by their character-unaware text encoder. Motivated by this, UDiffText [29] replaces the original CLIP text encoder in Stable Diffusion with a character-level text encoder with a unitary font. It also employs local attention loss for position control, relying on ground truth character segmentation maps.

However, these methods still encounter challenges in accurately rendering text within complex scenes and synthesizing characters in their ideal regions, attributed to their constrained character representation domain and lack of effective guidance for the model to autonomously estimate character-generated position, respectively.

## 3. Method

Given an input image $z$, a text region specified by a binary mask $\boldsymbol{B}$ and a text condition $c$, scene text synthesis is designed to render the given text onto the input image at the given text region. A natural idea for scene text synthesis is to formulate it into a latent diffusion process [15], which is equivalent to solving the following optimization problem:

$$\min_{(\theta,\vartheta)} \mathcal{L}_{LDM} \triangleq \mathbb{E}_{z,c,\epsilon\sim N(0,1),t} \parallel \epsilon - \epsilon_\theta(z_t, t, \psi_\vartheta(c), \boldsymbol{B}) \parallel_2^2,$$

where $\psi_\vartheta$ is the learnable character-level text encoder initialized by UDifftext [29] and $\theta$ is the model parameter vector. However, we argue that such a straightforward formulation could be problematic due to the following reasons:

- Without explicit guidance for rectifying characters' attention, character repetition and absence can inevitably occur in the rendered results as shown in Fig. 2.
- This formulation uniformly measures the mean distance between all pixels, without emphasizing the text regions. Consequently, text-unrelated regions can adversely affect the learning of the text encoder during backpropagation. The experimental analysis is provided in Sec. 4.3.
- Due to the intricate interactions among the U-Net and text encoder, end-to-end training is particularly challenging as shown in Sec. 4.3. Therefore, a more refined guidance for the training process is necessary.

In this paper, we propose DreamText and develop a heuristic iterative optimization pipeline to enable efficient training. Our key idea is to explicitly model the potential character positions in attention maps by encoding them into latent character masks in each step. These masks are then used to enhance the learning of the text encoder through our carefully crafted losses, facilitating the calibration of character attention in the subsequent step. This heuristic iterative process allows our model to dynamically alternate between optimizing character embedding and re-estimating character generation positions. The details are presented in the following sections.

### 3.1. Latent Diffusion with Refined Guidance

#### 3.1.1. Latent Character Mask

We utilize cross-attention maps to generate latent character masks, extracting characters' latent position information in the current step. Given latent image $z_t$ and text embedding $\psi_\vartheta(c)$, the cross-attention map in layer $l$ is defined as follows:


Figure 6. Qualitative comparative results against state-of-the-art methods.

$$\boldsymbol{Q}_l = z_t \boldsymbol{W}_l^q, \quad \boldsymbol{K}_l = \psi_\vartheta(c)\boldsymbol{W}_l^k, \tag{1}$$

$$\boldsymbol{A}_l = \text{softmax}\left(\frac{\boldsymbol{Q}_l \boldsymbol{K}_l^T}{\sqrt{d}}\right), \tag{2}$$

where $\boldsymbol{W}_l^q$ and $\boldsymbol{W}_l^k$ are learnable parameters and $d$ is the embedding dimension. The attention maps $\boldsymbol{A}_l$ is reshaped to $N \times H \times W$, where $N$ represents the number of text tokens such that each slice $\boldsymbol{A}_l^i \in \mathbb{R}^{H \times W}$ represents the region attended by the $i$-th token. These attention maps are then averaged across all layers of the U-Net to get the mean response $\bar{\boldsymbol{A}} = \frac{1}{L}\sum_{l=1}^{L} \boldsymbol{A}_l$ for the regions attended by character tokens.

To obtain the latent character masks $\boldsymbol{M} \in \mathbb{R}^{N \times H \times W}$, we first apply Gaussian blur to the attention map to execute low-pass filtering $\text{blur}(\bar{\boldsymbol{A}})$. This step helps mitigate excessive variance in the attended regions, ensuring a more uniform distribution of attention across relevant character regions. Subsequently, we employ a straightforward thresholding process to convert the blurred attention map into binary character masks. This process is encapsulated by the function $f(\cdot)$, which applies a threshold and assigns a value of 1 to pixel values exceeding the threshold, and 0 otherwise. The threshold is calculated as the mean value plus twice the variance of averaged attention maps in this work. That is:

$$f(\boldsymbol{X}) = \begin{cases} 1, \text{ if } x_{i,j} > \text{mean}(\boldsymbol{X}) + 2\text{std}(\boldsymbol{X}) \\ 0, \text{otherwise} \end{cases} . \tag{3}$$

Therefore, the equation for obtaining $\boldsymbol{M}$ can be expressed as:

$$\boldsymbol{M} = f(\text{blur}(\bar{\boldsymbol{A}})). \tag{4}$$

Based on the latent character masks $\boldsymbol{M}$, we employ several loss functions to optimize both the text encoder and U-Net iteratively. We will delve into these in the following sections.

#### 3.1.2. Masked Diffusion Loss

We first extend the Eq. 1 to highlight all desired character tokens in the current step. Specifically, within a text $c$ containing $k$ tokens of interest, the diffusion loss of the corresponding pixels obtained from each token's latent character mask is applied with an additional weighting factor $\gamma$. Formally, considering $\boldsymbol{M}_k = \bigvee_{i=1}^{k} \boldsymbol{M}_i$ to the union of the pixels of $k$ characters, the masked diffusion loss is formulated as,

$$\mathcal{L}_{mask} = \mathbb{E}_{z,c,\epsilon\sim N(0,1),t} \parallel (1+\gamma \boldsymbol{M}_k)(\boldsymbol{\epsilon} - \epsilon_\theta(z_t, t, \psi_\vartheta(c))) \parallel_2^2 .$$

We omit binary mask $\boldsymbol{B}$ here for the simplicity.

#### 3.1.3. Cross Attention Loss

The masked diffusion loss is designed to focus on all desired concept tokens within a prompt. Additionally, to ensure that each token encodes information specific to its corresponding synthesis position, we incorporate the cross-attention loss [2]. This loss encourages tokens to attend exclusively to their corresponding target regions,

$$\mathcal{L}_{attn} = \mathbb{E}_{z,c,t} \parallel C_{attn}(z_t, \psi_\vartheta(c)_i) - \boldsymbol{M}_i \parallel_2^2 . \tag{5}$$

Here, $C_{attn}(z_t, \psi_\vartheta(c)_i)$ represents the cross-attention map between the visual representation $z_t$ and the token $\psi_\vartheta(c)_i$, while $\boldsymbol{M}_i$ denotes the latent character mask of $\psi_\vartheta(c)_i$.

However, the losses mentioned above alone are inadequate for achieving optimal character representation. This is primarily due to the inherent noise in the latent character masks, which frequently results in under- or over-segmentation of the ideal character regions during training. To mitigate the influence of noise and prevent skewed learning of character representation, we introduce additional losses [29] as shown below.

Table 1. Quantitative comparison against baselines.

| Methods | *SeqAcc-Recon* | | | | *SeqAcc-Editing* | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ICDAR13(8ch) | ICDAR13 | TextSeg | LAION-OCR | ICDAR13(8ch) | ICDAR13 | TextSeg | LAION-OCR | FID | LPIPS |
| CVPR'22 SD-Inpainting [15] | 0.32 | 0.29 | 0.11 | 0.15 | 0.08 | 0.07 | 0.04 | 0.05 | 26.78 | 0.0696 |
| arXiv'23 DiffSTE [8] | 0.45 | 0.37 | 0.50 | 0.41 | 0.34 | 0.29 | 0.47 | 0.27 | 51.67 | 0.1050 |
| AAAI'23 MOSTEL [14] | 0.75 | 0.68 | 0.64 | 0.71 | 0.35 | 0.28 | 0.25 | 0.44 | 25.09 | 0.0605 |
| NIPS'23 TextDiffuser [5] | 0.87 | 0.81 | 0.68 | 0.80 | 0.82 | 0.75 | 0.66 | 0.64 | 32.25 | 0.0834 |
| ICLR'24 AnyText [19] | 0.89 | 0.87 | 0.81 | 0.86 | 0.81 | 0.79 | 0.80 | 0.72 | 22.73 | 0.0651 |
| ECCV'24 UDiffText [29] | 0.94 | 0.91 | 0.93 | 0.90 | 0.84 | 0.83 | 0.84 | 0.78 | 15.79 | 0.0564 |
| **DreamText** | **0.95** | **0.94** | **0.96** | **0.93** | **0.87** | **0.89** | **0.91** | **0.88** | **12.13** | **0.0328** |

#### 3.1.4. Cross-modal Aligned Loss

To obtain accurate and robust embeddings, we introduce an image encoder $\xi$, text head $H_t$, and visual head $H_v$ to align cross-modal character features. Specifically, we compute the cosine similarity objective between visual and text representations to maximize the alignment between two modalities:

$$\mathcal{L}_{align} = \frac{\langle H_t(\boldsymbol{y}), H_v(\xi(\boldsymbol{I}))\rangle}{\|H_t(\boldsymbol{y})\|_2 \cdot \|H_v(\xi(\boldsymbol{I}))\|_2}. \tag{6}$$

where $\boldsymbol{I}$ represents the text image segmented via the bounding box from datasets and $\langle , \rangle$ presents the inner product of two vectors. Note that to alleviate noise stemming from background and color variations, we preprocess the text image by converting it to grayscale.

Through training, the text head learns to map the whole information of multiple characters into the text feature space. Similarly, the image head maps the overall visual text information into the visual feature space. This effectively ensures the overall textual and visual representations correspond, capturing high-level cross-modal consistency.

#### 3.1.5. Character Id Loss

Additionally, we introduce character Id loss to ensure that the learned embeddings are highly distinguishable. To be precise, a multi-label classification head $H_l$ is introduced to predict character indices from text embeddings $\boldsymbol{y}$. Then, the cross-entropy objective is formulated,

$$\mathcal{L}_{id} = -\sum_{i=1}^{N}\sum_{j=1}^{K} \boldsymbol{l}_{i,j} \log(H_l(\boldsymbol{y})_j), \tag{7}$$

where $N$ and $K$ represent the number of characters and the possible indices respectively while $\boldsymbol{l}$ denotes the ground truth labels. This loss aggregates over all characters in the target text, ensuring the text encoder produces distinguishable embeddings for each character.

The complete objective of our training strategy can be expressed as

$$\mathcal{L} = \mathcal{L}_{mask} + \alpha\mathcal{L}_{attn} + \beta(\mathcal{L}_{align} + \mathcal{L}_{id}). \tag{8}$$

### 3.2. Optimization Strategy

#### 3.2.1. Heuristic Alternate Optimization

Our objective involves numerous discrete variables, rendering it non-differentiable and making vanilla SGD invalid. Thus, we employ a heuristic alternate optimization strategy, utilizing latent character masks that encapsulate potential synthesis position information, to facilitate a symbiotic relationship between text encoder and U-Net. Specifically, we perform an alternating update between the tokens given the latent character masks and the masks themselves. During optimization, the masks are computed by Eq. 4, while for other parameters, we fix the masks and calculate gradients accordingly. Using the losses above, we first optimize the representation of specific characters given masks in each iteration, which, in turn, enables the generator to rectify the character's attention in subsequent steps. This process allows model to dynamically alternate between optimizing character embeddings and re-estimating character masks.

#### 3.2.2. Balanced Supervision on Character Attention

The initial training phase may challenge the generator in directing attention to the intended generation region for each character. Unlike existing methods that excessively constrain character attention which inevitably limits the model's flexibility, we adopt a balanced supervision. Initially, we guide attention calibration by cross-entropy objective between latent character masks and character segmentation masks from datasets similar to Eq. 7 for a warm-up in the earlier stage. Once the model preliminarily acquires the capability to estimate ideal generation positions (approximately 25k steps as shown in Fig. 4), we cease guidance, enabling it to engage in autonomous iterative learning. This strategy balances constraining the model and unleashing its flexibility in estimating characters' optimal generation positions.

## 4. Experiments

### 4.1. Implementation Details

#### 4.1.1. Datasets

Several datasets are used in our experiments: (1) **SynthText** [6] comprises 800,000 images containing around 8 million synthetic word instances. Each text instance is annotated with its corresponding text string, along with word-level and character-level bounding boxes. (2) **LAION-OCR** [5] contains 9,194,613 filtered high-quality text images including advertisements, notes, posters, covers, memes, logos, etc. (3) **ICDAR13** [9] is widely recognized as the benchmark dataset for evaluating near-horizontal text detection, which consists

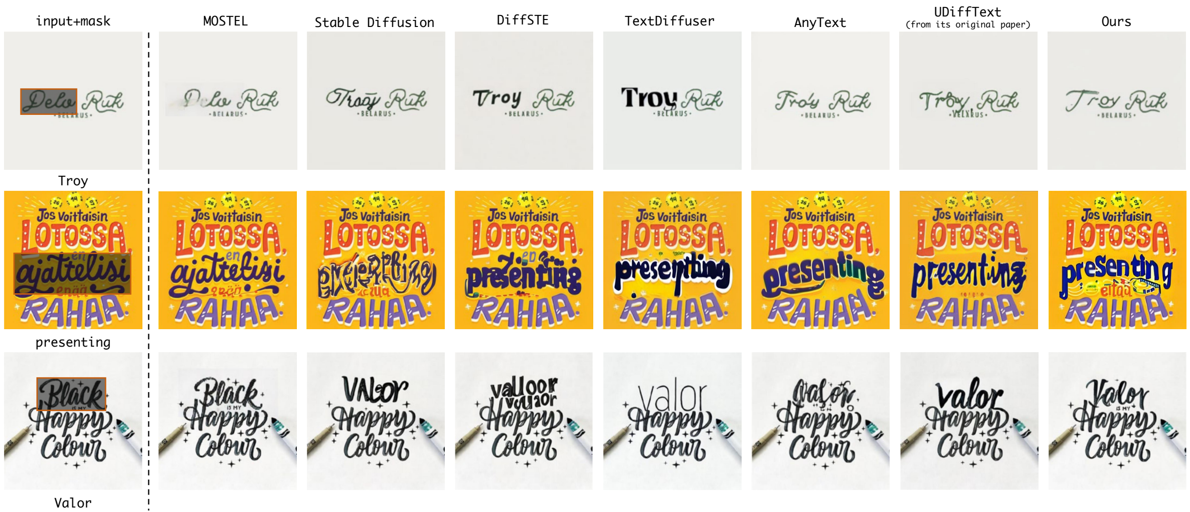


Figure 7. Qualitative comparative results against state-of-the-art methods.

of 233 test images used for evaluation purposes. (4) **TextSeg** [26] comprises 4,024 real-world text images sourced from various sources such as posters, greeting cards, book covers, logos, road signs, billboards, digital designs, handwritten notes, and more. Note that only SynthText and LAION-OCR datasets provide the character segmentation maps. We train our model using the training subsets of these datasets and randomly select 100 images from test subsets for testing.

#### 4.1.2. Training Configurations

We train our model based on the pre-trained checkpoint of SD-v2.0 inpainting version and text encoder in [29]. The pre-trained Vision Transformer [1] is used as our image encoder. The model is fine-tuned using 4 NVIDIA A100 GPUs on LAION-OCR for 200k steps, SynthText for 150k steps, TextSeg for 50k steps, and ICDAR13 for an additional 10k steps. We set $\alpha$ to 0.01, and $\beta$ to 0.001. Additionally, we utilize a batch size of 16 and a learning rate of $5 \times 10^{-5}$. The inference time of our DreamText is 8.5 seconds only.

#### 4.1.3. Evaluation

**Text accuracy**: We assess all methods on two tasks: scene text reconstruction and scene text editing [29]. For the scene text reconstruction task, we use the models to reconstruct the text image using the provided ground truth text label and binary mask. In the scene text editing task, we replace the original text in each image with a random word and evaluate the models by generating images containing the edited text. We use an off-the-shelf scene text recognition (STR) model [3] to identify the rendered text and then evaluate word-level correctness using sequence accuracy (SeqAcc) by comparing the STR result with the ground truth.

**Image quality**: we utilize Fréchet Inception Distance (FID) [7] to quantify the distance between the text images in the dataset and the generated images. Additionally, the Learned

Table 2. Accumulated results of losses.

| Setting | *Average SeqAcc* | | | |
|---|---|---|---|---|
| | Recon | Editing | FID | LPIPS |
| Base | 0.218 | 0.060 | 26.78 | 0.0696 |
| $+\mathcal{L}_{mask}$ | 0.425 | 0.259 | 23.21 | 0.0528 |
| $+\mathcal{L}_{attn}$ | 0.698 | 0.532 | 19.72 | 0.0483 |
| $+\mathcal{L}_{align}$ | 0.884 | 0.801 | 15.41 | 0.0392 |
| $+\mathcal{L}_{id}$ | **0.940** | **0.887** | **12.13** | **0.0328** |

Perceptual Image Patch Similarity (LPIPS) [28] is utilized as an additional metric.

### 4.2. Results

**Quantitatively**: In Tab. 6, we present a quantitative analysis comparing DreamText against baseline methods. Our method demonstrates a significant advantage across all quantitative metrics, highlighting the exceptional visual quality of the scene text images it generates. Notably, when compared to the previous state-of-the-art model UDiffText, DreamText outperforms it by 3.66 in terms of FID. While UDiffText and AnyText excel in image quality compared to other methods, except DreamText, they tend to perform less satisfactorily in terms of sequence accuracy, especially in text editing. This discrepancy may be attributed to its restricted representation domain and deflected attention, resulting in less accurate text rendering within complex scenes.

**Qualitatively**: As depicted in Fig. 6, our method outperforms other baselines in synthesizing more coherent and accurate text within polystylistic scenes. For example, in Fig. 6 (3rd row), MOSTEL and Stable Diffusion fail to generate the text, while TextDiffuser and UDiffText encounter the issue of character absence. Similarly, in Fig. 6 (1st row), all baselines fail to generate the text on the provided poster. Moreover, we further compare our method with the latest ap-

Table 3. Hyperparameter analysis.

| $\alpha$ ($\beta = 0$) | *Average SeqAcc* Recon | Editing | $\beta$ ($\alpha$=0.01) | *Average SeqAcc* Recon | Editing |
|---|---|---|---|---|---|
| 1 | 0.672 | 0.558 | 1 | 0.864 | 0.766 |
| 0.1 | 0.724 | 0.615 | 0.1 | 0.896 | 0.817 |
| **0.01** | 0.748 | **0.623** | 0.01 | 0.921 | 0.864 |
| 0.001 | **0.753** | 0.604 | **0.001** | **0.940** | **0.887** |

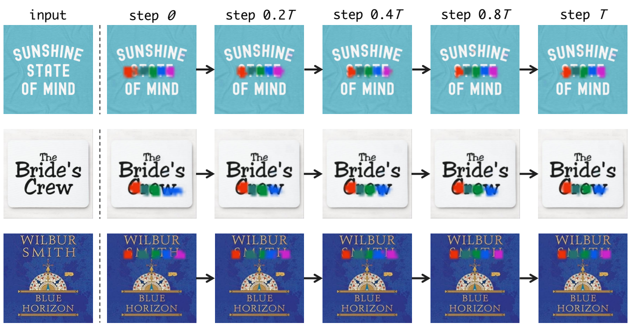


Figure 8. The visualized attention results of all characters across several steps during training.

proach, UDiffText, using the samples provided in its original paper for additional qualitative evaluation. The visualization results are presented in Fig. 7, offering additional evidence of our method's superiority in generating high-fidelity text.

### 4.3. Ablation Studies and Effectiveness Analysis

**Ablation on Losses**: We employ SD-v2.0 inpainting version as the base method. As shown in Tab. 2, we observe a significant drop in performance across all metrics when these losses are removed. Additionally, these results highlight that the masked diffusion loss and cross-attention loss alone are insufficient for achieving satisfactory character representation. This may be attributed to the inherent noise in the latent character masks. To sum up, these ablation results underscore the effectiveness of each objective within our heuristic alternate optimization strategy in improving various aspects of synthesized scene texts.

**Choice of $\alpha$ and $\beta$**: In our experiments, we vary the values of $\alpha$ and $\beta$ within the range [0.001, 0.01, 0.1, 1]. The results are presented in Tab. 3. Initially, we focus on exploring the impact of $\alpha$ while setting $\beta = 0$. We find that the optimal performance in text reconstruction occurs when $\alpha = 0.001$. However, this choice yields poor results in editing tasks. Consequently, we opt for a more balanced approach, selecting $\alpha = 0.01$. Subsequently, with $\alpha$ fixed at 0.01, we investigate the effect of $\beta$. We find that the optimal performance is achieved when $\beta = 0.001$.

**Effectiveness of Heuristic Alternate Optimization**: We analyze the efficacy of our strategy by visualizing the evolution of character attention maps across several training steps. As illustrated in Fig. 8, the model exhibits deflected attention

Table 4. Ablation study results on warm-up steps.

| | *Average SeqAcc* Recon ↑ | Editing ↑ | FID ↓ | mIoU ↑ |
|---|---|---|---|---|
| 15k | 0.884 | 0.852 | 13.82 | 0.681 |
| 20k | 0.913 | 0.873 | 13.38 | 0.692 |
| **25k** | **0.940** | 0.887 | **12.13** | **0.722** |
| 30k | 0.921 | **0.891** | 13.24 | 0.703 |

initially but progressively corrects it in subsequent training steps. Ultimately, attention becomes concentrated on the desired positions for all characters, indicating the effectiveness of our approach. Besides, Fig. 4 also illustrates the efficacy of this strategy in continuously improving the generator's ability to estimate the optimal characters' position by autonomously rectifying the attention during training.

**Choice of Warm-up Steps**: We conduct an ablation experiment to analyze the impact of different warm-up steps on model performance. The results, presented in Tab. 4, demonstrate how varying the number of warm-up steps affects the performance. As the number of warm-up steps increases from 15k to 25k, we observe consistent improvements across all metrics. However, at 30k steps, while the editing accuracy slightly improves, the FID score worsens, suggesting a potential trade-off between image fidelity and editing capability. Overall, these findings highlight that 25k warm-up steps strike the best balance between reconstruction accuracy, editing performance, and image quality, making it the optimal choice for our method.

## 5. Conclusion

This paper presents DREAMTEXT, a novel approach for high-fidelity scene text synthesis. We reconstruct the diffusion training process to expose and rectify the model's attention at the character level while strengthening its learning of text regions. This inevitably involves a hybrid optimization challenge, combining discrete and continuous variables, which we effectively address through our heuristic alternate optimization strategy. Meanwhile, we jointly train the text encoder and generator to comprehensively learn and utilize diverse fonts present in the training dataset, fostering a synergistic relationship between learning character embeddings and re-estimating character attention. Both qualitative and quantitative results highlight the superiority of our method.

## 6. Acknowledgments

This work was supported by the National Nature Science Foundation of China (62472097), AI for Science Foundation of Fudan University (FudanX24AI028), and Shanghai Innovation Institute. The computations in this research were performed on the CFFF platform of Fudan University. Cheng Jin is the corresponding author of this work.

# References


[1] Rowel Atienza. Vision transformer for fast and efficient scene text recognition. In *ICDAR*, pages 319–334, 2021. 7

[2] Omri Avrahami, Kfir Aberman, Ohad Fried, Daniel Cohen-Or, and Dani Lischinski. Break-a-scene: Extracting multiple concepts from a single image. In *SIGGRAPH Asia*, pages 1–12, 2023. 5

[3] Darwin Bautista and Rowel Atienza. Scene text recognition with permuted autoregressive sequence models. In *ECCV*, pages 178–196, 2022. 7

[4] Haoxing Chen, Zhuoer Xu, Zhangxuan Gu, Yaohui Li, Changhua Meng, Huijia Zhu, Weiqiang Wang, et al. Diffute: Universal text editing diffusion model. In *NeurIPS*, 2024. 3

[5] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser: Diffusion models as text painters. In *NeurIPS*, 2023. 2, 3, 6

[6] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, pages 2315–2324, 2016. 6

[7] 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. In *NeurIPS*, 2017. 7

[8] 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. 2, 3, 6

[9] 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 *ICDAR*, pages 1484–1493, 2013. 6

[10] Yuxin Kong, Canjie Luo, Weihong Ma, Qiyuan Zhu, Shenggao Zhu, Nicholas Yuan, and Lianwen Jin. Look closer to supervise better: One-shot font generation via component-based discriminator. In *CVPR*, pages 13482–13491, 2022. 3

[11] Sanyam Lakhanpal, Shivang Chopra, Vinija Jain, Aman Chadha, and Man Luo. Refining text-to-image generation: Towards accurate training-free glyph-enhanced image generation. *arXiv preprint arXiv:2403.16422*, 2024. 2

[12] Zeyu Liu, Weicong Liang, Zhanhao Liang, Chong Luo, Ji Li, Gao Huang, and Yuhui Yuan. Glyph-byt5: A customized text encoder for accurate visual text rendering. *arXiv preprint arXiv:2403.09622*, 2024. 2

[13] Song Park, Sanghyuk Chun, Junbum Cha, Bado Lee, and Hyunjung Shim. Multiple heads are better than one: Few-shot font generation with multiple localized experts. In *ICCV*, pages 13900–13909, 2021. 1

[14] 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, 3, 6

[15] 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. 4, 6

[16] Prasun Roy, Saumik Bhattacharya, Subhankar Ghosh, and Umapada Pal. Stefann: scene text editor using font adaptive neural network. In *CVPR*, pages 13228–13237, 2020. 1, 3

[17] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, pages 36479–36494, 2022. 2

[18] Wataru Shimoda, Daichi Haraguchi, Seiichi Uchida, and Kota Yamaguchi. De-rendering stylized texts. In *ICCV*, pages 1076–1085, 2021. 3

[19] Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Geng, and Xuansong Xie. Anytext: Multilingual visual text generation and editing. *ICLR*, 2024. 6

[20] Yibin Wang, Yuchao Feng, Jie Wu, Honghui Xu, and Jianwei Zheng. Ca-gan: Object placement via coalescing attention based generative adversarial network. In *ICME*, pages 2375–2380. IEEE, 2023. 3

[21] Yibin Wang, Honghui Xu, Changhai Zhou, Weizhong Zhang, and Cheng Jin. Enhancing object coherence in layout-to-image synthesis. *arXiv preprint arXiv:2311.10522*, 2023. 2

[22] Yibin Wang, Weizhong Zhang, and Cheng Jin. Magicface: Training-free universal-style human image customized synthesis. *arXiv preprint arXiv:2408.07433*, 2024. 2

[23] Yibin Wang, Weizhong Zhang, Jianwei Zheng, and Cheng Jin. High-fidelity person-centric subject-to-image synthesis. In *CVPR*, 2024. 2

[24] Yibin Wang, Weizhong Zhang, Jianwei Zheng, and Cheng Jin. Primecomposer: Faster progressively combined diffusion for image composition with attention steering. *arXiv preprint arXiv:2403.05053*, 2024. 2

[25] Liang Wu, Chengquan Zhang, Jiaming Liu, Junyu Han, Jingtuo Liu, Errui Ding, and Xiang Bai. Editing text in the wild. In *ACM MM*, pages 1500–1508, 2019. 3

[26] Xingqian Xu, Zhifei Zhang, Zhaowen Wang, Brian Price, Zhonghao Wang, and Humphrey Shi. Rethinking text segmentation: A novel dataset and a text-specific refinement approach. In *CVPR*, pages 12045–12055, 2021. 7

[27] Qiangpeng Yang, Jun Huang, and Wei Lin. Swaptext: Image based texts transfer in scenes. In *CVPR*, pages 14700–14709, 2020. 1, 3

[28] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 7

[29] Yiming Zhao and Zhouhui Lian. Udifftext: A unified framework for high-quality text synthesis in arbitrary images via character-aware diffusion models. In *ECCV*, 2024. 2, 3, 4, 5, 6, 7