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

Abstract. Text-to-Image (T2I) generation based on diffusion models has garnered significant attention in the last few years. Although these image synthesis methods produce visually appealing results, they frequently exhibit spelling errors when rendering text within the generated images. Such errors manifest as missing, incorrect or extraneous characters, thereby severely constraining the performance of text image generation based on diffusion models. To address the aforementioned issue, this paper proposes a novel approach for text image generation, utilizing a pre-trained diffusion model (i.e., Stable Diffusion). Our approach involves the design and training of a light-weight character-level text encoder, which replaces the original CLIP encoder and provides more robust text embeddings as conditional guidance. Then, we fine-tune the diffusion model using a large-scale dataset, incorporating local attention control under the supervision of character-level segmentation maps. Finally, by employing an inference stage refinement process, we achieve a notably high sequence accuracy when synthesizing text in arbitrarily given images. Both qualitative and quantitative results demonstrate the superiority of our method to the state of the art. Furthermore, we showcase several potential applications of the proposed UDiffText, including text-centric image synthesis, scene text inpainting, etc. Our code and model are available at https://github.com/ZYM-PKU/UDiffText.

Keywords: Diffusion Model · Text Image Synthesis

1 Introduction

Since the proposal of denoising diffusion probability model (DDPM) [14], it has shown great potential in the field of image generation. In comparison with traditional generative adversarial networks (GANs) [10], this kind of hidden-variable probabilistic graphical model has significant advantages, which are specifically reflected in its simple optimization objectives and clear iterative definition of the generation process. Especially, it does not suffer the problem of loss convergence difficulty when the model parameters are expanded. With the evolution

* Corresponding author

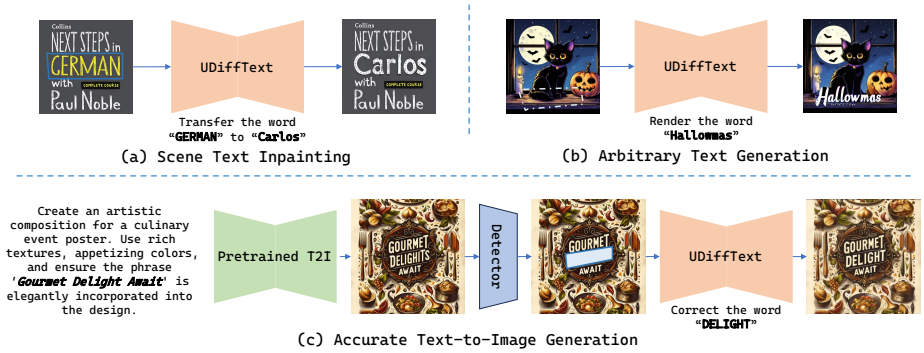

Fig. 1: The proposed UDiffText is capable of synthesizing accurate and harmonious text in either synthetic or real-word images, thus can be applied to tasks like scene text inpainting (a), arbitrary text generation (b) and accurate T2I generation (c).

of multimodal approaches, the integration of textual guidance into the diffusion model has given rise to large T2I generation models [3, 25, 27, 28, 31]. The majority of these models have a substantial number of parameters and are trained on extremely large-scale text-image pair datasets, typically in the billion-level range. Their capability of producing high-fidelity images with straightforward text prompts facilitates their seamless adaptation to a range of generative tasks.

Although T2I generation models have made significant strides and can automate the process of artistic visual design to some extent, they still exhibit numerous limitations. For instance, when generating images that include human figures, these models often produce inaccurate or missing details in hands and faces. When synthesizing images with the desired text, these models often encounter serious spelling issues including incorrect, missing or repetitive characters. In some cases, they fail entirely to render text in generated images. Some researchers [21] pointed out that these text rendering issues primarily stem from the inadequate information provided by the text encoder. They suggested that incorporating a character-aware text encoder with a large number of parameters could mitigate this problem to some extent. The authors of DALL-E 3 [3] also noted a limitation when the model encounters quoted text in a prompt: the T5 text encoder they utilize actually interprets tokens representing whole words and must map these to letters in an image, inevitably leading to unstable text rendering.

We suspect that those spelling issues in text synthesis is closely linked to the fundamental problems of existing T2I models, namely catastrophic neglect and incorrect attribute binding. To address this problem, we adopt and train a light-weight character-level text encoder to replace the original CLIP encoder employed in Stable Diffusion [28], thus providing more robust conditional guidance for the diffusion model. We then fine-tune a small portion of the model using the denoising score matching scheme and a proposed local attention map constraint. Finally, after implementing a refinement process during the inference

stage, we shape the diffusion model into a powerful text designer capable of rendering precise words in images. Consequently, it can be utilized to precisely synthesize arbitrary text in given images based solely on text conditions. We summarize our main contributions as follows:

- We propose a diffusion model-based text image synthesis method, UDiffText, to address the text rendering challenges of existing T2I models. We leverages a character-level text encoder to derive robust text embeddings and employs a combination of the local attention loss and the scene text recognition loss to train our model on large-scale datasets.
- The incorporation of segmentation map supervision offers a novel training strategy for T2I models, leading to enhanced text rendering performance. Experimental results demonstrate the effectiveness and superiority of our proposed method to the state of the art in terms of both text rendering accuracy and visual context coherency.
- As shown in Fig. 1, we demonstrate several potential applications of our proposed UDiffText, including scene text inpainting, arbitrary text generation as well as T2I generation with precise text content.

2 Related Work

2.1 Image Synthesis with Diffusion Models

Recent state-of-the-art methods in image synthesis mostly utilize diffusion models (DMs). Ever since the introduction of denoising diffusion probability model (DDPM) [14], large T2I models [3, 25, 27, 28, 31] have achieved significant advancements in high-resolution image synthesis, exhibiting considerable diversity. Our research is conducted on the basis of Stable Diffusion [28] and relevant efficient sampling algorithms [18, 33].

2.2 Guided Diffusion

While the advent of classifier-free guidance [15] has enhanced the generation performance of diffusion models, numerous methods have been explored to control these models using conditions from different modalities. Some approaches [4, 24, 30] concatenate image conditions with noised latent variables as model input to furnish visual information. Others [9, 20] utilize prompt tuning for concept-specific generation. Besides, certain methods [23, 42] construct bypass network to control diffusion models using flexible pixel-domain conditions.

Notably, it is widely accepted that the cross-attention (CA) mechanism is pivotal in the generation process. Prompt-to-prompt [12] evidences that CA maps are instrumental in determining the spatial layout of objects in generated images. Perfusion [34] elaborates that the "Keys" in the CA mechanism govern the region of objects, while the "Values" dictate the features incorporated into the region. Structured Diffusion [8] employs noun phrase extraction to obtain more accurate CA features, thereby mitigating semantic attribute leakage.

FastComposer [36] aligns CA maps with subject segmentation masks to address the problem of identity blending in multi-subject image generation. Attend-and-excite [5] directs diffusion models to refine the CA units to attend to all subject tokens in the text prompt, thus alleviating the issue of catastrophic neglect. In this study, we attempt to constraint the CA maps of our diffusion model under the guidance of character-level segmentation maps to gain better performance.

2.3 Scene Text Synthesis

GAN-based scene text editing methods exhibit proficiency in generating coherent text within a specific visual context. Textstylebrush [19] utilizes a StyleGAN architecture to generate new content aligned with the source style. STEFANN [29] constructs a FANnet to edit a single character and implements a placement algorithm to generate the expected word. SRNet [35], MOSTEL [26] and Swap-Text [39] divide the task into two primary parts: background inpainting and text style transfer. This division facilitates whole word editing in an end-to-end manner. Despite their simplicity and effectiveness, the capacity of these methods to generate high-resolution and polystylistic text images remains limited.

More recently, a number of approaches that aim to tackle the aforementioned text rendering challenges associated with diffusion models have been proposed. They leverage the robust capabilities of DMs to synthesis scene text, thereby enhancing the quality and variety of the generated content. DiffSTE [16] uses the dual encoder structure (character text encoder and instruction text encoder) and performs instruction tuning to provide more accurate control for the backbone network. DiffUTE [6] uses an OCR-based glyph encoder to obtain glyph guidance from the rendered glyph image. Similarly, GlyphDraw [22] leverages an additional image encoder and a fusion module to inject glyph condition and the fine-tuned model is able to generate images with coherent Chinese text. Glyph-Control [40] applies ControlNet [42] to text image generation tasks by using the rendered reference image as both position and glyph guidance. TextDiffuser [7] chooses to concatenate the segmentation mask as conditional input and uses the character-aware loss to control the generated characters more precisely. In this study, we supplant the original CLIP text encoder in Stable Diffusion with a more robust character-level text encoder. This substitution equips the CA module with expressive and highly distinguishable character-aware embeddings. We firstly employ contrastive learning under visual supervision from a well-trained scene text recognition (STR) model to train the encoder. Then we fine-tune the CA blocks to yield more efficient CA "Keys" and "Values", which help the model generate more accurate text images.

3 Method

As mentioned above, we aim to design a unified framework for high-quality text synthesis in both synthetic and real-world images. The proposed method, UDiffText, is built based on the inpainting variant of Stable Diffusion (v2.0).

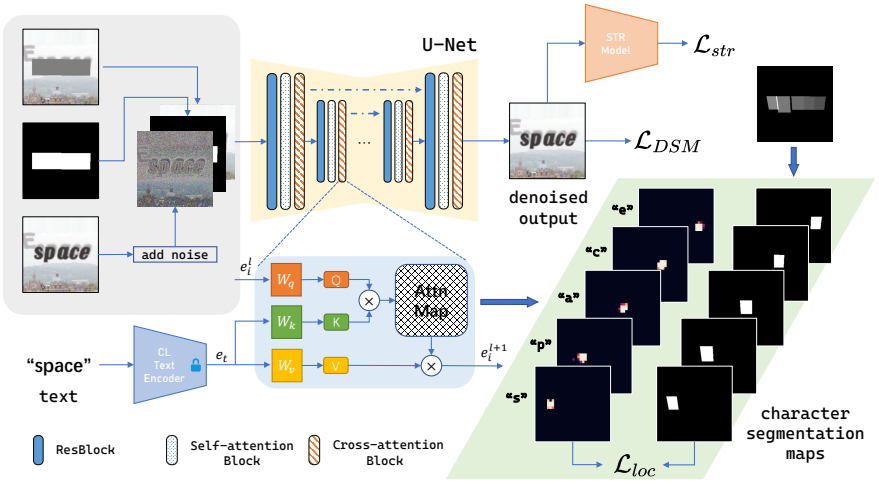

Fig. 2: An overview of the training process of our proposed UDiffText. We build our model based on the inpainting version of Stable Diffusion (v2.0). A character-level (CL) text encoder is utilized to obtain robust embeddings from the text to be rendered. We train the model using denoising score matching (DSM) together with the local attention loss calculated based on character-level segmentation maps and the auxiliary scene text recognition loss. Note that only the parameters of cross-attention (CA) blocks are updated during training.

An overview of our method is depicted in Fig. 2. Specifically, we first design and train a light-weight character-level (CL) text encoder as a substitute for the original CLIP text encoder. Then, we train the model using the denoising score matching (DSM) loss in conjunction with the local attention loss and scene text recognition loss. More details of our proposed UDiffText will be elaborated in the following subsections.

3.1 Character-level Text Encoder

As expounded in prior research [21], a character-aware text encoder is deemed crucial in rectifying the issue of spelling errors in T2I models. However, the CLIP text encoder and T5 encoder, which are prevalently employed in T2I models, do not tokenize prompts at the character level. This results in the backbone network perceiving the entire word (subword) rather than its internal structure. A potential substitute for these encoders could be pre-trained character-aware transformers like ByT5 [38]. However, only models with large amounts of parameters can exhibit reasonable performance, making the generation process inefficient and leading to unnecessary computational waste. A possible solution is to utilize encoders to obtain character-level embeddings with the help of pixel domain references. Yet, how to select appropriate references for the visual encoder is still an unsolved problem due to the requirement of a precise and generalized text representation to synthesize text images with diverse visual contexts.

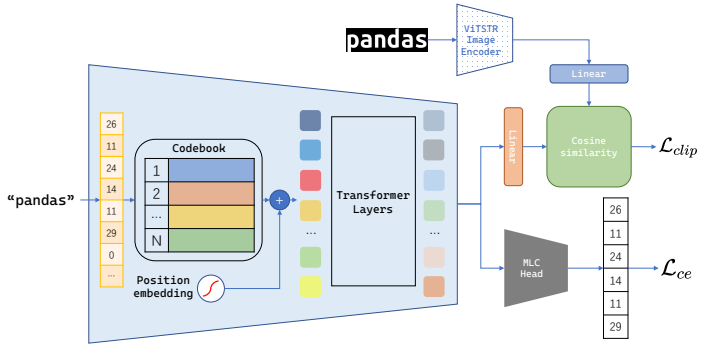

Fig. 3: The network architecture of our character-level text encoder. A codebook is employed to translate the character indices into a sequence of learnable embeddings. These embeddings are enhanced by position embeddings and then fed into a transformer to generate the encoded output.

In this research, we design a CLIP-like text encoder that processes words at the character level. As shown in Fig. 3, a target word is first mapped to corresponding indices and then converted into learnable embeddings using a codebook. Transformer layers are concatenated to produce the final output of shape (B, L, d_{emb}), where B represents the batch size, L indicates the maximum sequence length, and d_{emb} denotes the dimension of the encoder. To obtain robust generalized embeddings, we train the text encoder $\mathcal{E}text$ using a combination of the contrastive loss and the multi-label classification loss. We first render the target word with a standard font style into an image. Then we use the ViTSTR [1] model, a scene text recognizer, as the image encoder $\mathcal{E}image$ to obtain robust visual features. A multi-label classification head $\mathcal{H}_{MLC}$ is trained concurrently to predict character indices Ids given text embeddings. The calculation of the training loss is detailed in the following equations, where $\mathcal{T}$ and $\mathcal{I}_{\mathcal{T}}$ represent the text label and corresponding image, respectively, and W_t, W_i are linear mapping matrices. We employ a cosine similarity ($\boldsymbol{CS}$) objective to align cross-modal features and use cross-entropy ($\boldsymbol{CE}$) as a multi-label classification loss to ensure that the learned embeddings are highly distinguishable:

$$\mathbf{e}_{text} = \mathcal{E}_{text}(\mathcal{T}), \quad \mathbf{e}_{image} = \mathcal{E}_{image}(\mathcal{I}_{\mathcal{T}}), \tag{1}$$

$$\mathcal{L}_{clip} = -\boldsymbol{CS}(W_t\mathbf{e}_{text}, W_i\mathbf{e}_{image}), \tag{2}$$

$$\mathcal{L}_{ce} = \boldsymbol{CE}(\mathcal{H}_{MLC}(\mathbf{e}_{text}), Ids), \tag{3}$$

$$\mathcal{L} = \mathcal{L}_{clip} + \lambda_{ce}\mathcal{L}_{ce}. \tag{4}$$

3.2 Training Strategy

Our system is constructed based on the inpainting version of Stable Diffusion [28] (v2.0). During the training stage, the model functions as a denoiser, accepting a noised text image $\mathbf{x}_0 + \mathbf{n}$ of shape (H, W), a binary mask $\mathcal{M}$ of the text region and the masked image $\mathbf{x}_{\mathcal{M}} = (\boldsymbol{J} - \mathcal{M}) \odot \mathbf{x}_0$ as input ($\boldsymbol{J}$ is the all-ones matrix), and predicting the original text image as output. We utilize the denoising score

matching (DSM) loss to train a denoiser for the specific text rendering task with the text condition $\mathcal{T}$:

$$\mathcal{L}_{DSM} = \lambda_\sigma \left\| D_{\boldsymbol{\theta}} \left(\mathbf{x}_0 + \mathbf{n}; \sigma, \mathcal{T}, \mathcal{M}, \mathbf{x}_{\mathcal{M}} \right) - \mathbf{x}_0 \right\|_2^2, \tag{5}$$

where $D_{\boldsymbol{\theta}}$ is a U-Net denoiser with the learnable parameter $\boldsymbol{\theta}$. $(\mathbf{x}_0, \mathcal{T}, \mathcal{M}) \sim p_{\text{data}}$ represents the text image, text label and binary mask of text region which are randomly sampled from the dataset. $\mathbf{n} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \boldsymbol{I}_d\right)$ is the gaussian noise added to the text image and σ represents the noise level. We set $\lambda_\sigma = \sigma^{-2}$ as the weighting function.

Our experimental results indicate that the DSM loss alone is insufficient to empower the model to render accurate text in generated images. This is mainly due to the fact that the $L2$ distance merely measures the mean distance between pixels, rather than the accuracy of character representation. To address this problem, we incorporate a local attention loss to regulate the cross-attention maps of the model, a strategy similar to the approach adopted in [36].

As mentioned in Sec. 2.2, we expect the model to learn appropriate projection matrices in the cross-attention blocks. This enables the computed attention map to attend to corresponding character regions, and the learned character features could be appended to these regions. To achieve this goal, we utilize the supervision from character segmentation maps in our dataset. Specifically, for a character sequence $\mathcal{T} = \left\{\mathbf{c}^1, \mathbf{c}^2, \ldots \mathbf{c}^L\right\}$, its corresponding segmentation map can be denoted as $\mathcal{S}_T = \left\{\mathbf{S}^1, \mathbf{S}^2, \ldots \mathbf{S}^L\right\}$, where $\mathbf{S}^i$ of shape (H, W) is a binary mask of the corresponding character $\mathbf{c}^i$ in the image. We can derive the attention maps $\mathcal{A}_i$ from each cross-attention block i of the U-Net:

$$\mathcal{Q}_i = W_i^Q \mathbf{e}_{image},\ \mathcal{K}_i = W_i^K \mathbf{e}_{text},\ \mathcal{V}_i = W_i^V \mathbf{e}_{text}, \tag{6}$$

$$\mathcal{A}_i = \boldsymbol{softmax}\left(\mathcal{Q}_i \mathcal{K}_i^T / \sqrt{d}\right) \mathcal{V}_i. \tag{7}$$

In this step, we partition the attention maps $\mathcal{A}_i$ on the dimension of sequence length into $\mathcal{A}_i = \left\{\mathbf{A}_i^1, \mathbf{A}_i^2, \ldots \mathbf{A}_i^L\right\}$. Each $\mathbf{A}_i^j$ of shape (H, W) reflects the region of interest (ROI) of block $\mathbf{b}_i$ on the character $\mathbf{c}^j$. Subsequently, the local attention loss can be computed as follows:

$$\begin{aligned} \mathcal{L}_{loc} = \frac{1}{C} \sum_{i=1}^{C} \Bigg\{ & \frac{1}{L} \sum_{j=1}^{L} \left(\boldsymbol{max}\left(\mathbb{G}\left(\mathbf{A}_i^j\right) \odot \left(\boldsymbol{J} - \mathbf{S}^j\right)\right)\right) \\ & - \frac{1}{L} \sum_{j=1}^{L} \left(\boldsymbol{max}\left(\mathbb{G}\left(\mathbf{A}_i^j\right) \odot \mathbf{S}^j\right)\right) \Bigg\}, \end{aligned} \tag{8}$$

where C represents the number of cross-attention blocks in the U-Net, $\mathbb{G}$ denotes a Gaussian blur and $\odot$ means the Hadamard product. The Gaussian blur is employed to perform low-pass filtering on the attention map, which helps to prevent excessive variance in the attended region. This approach ensures that the attention is distributed more evenly across the relevant regions, contributing to more accurate and stable model performance.

To enhance the text rendering accuracy, we incorporate the scene text recognition (STR) loss. Specifically, we employ a pre-trained STR model [2] to operate

on the text region in the denoised results, and apply cross-entropy ($\boldsymbol{CE}$) to measure the correctness of the rendered word:

$$\mathcal{L}_{str} = \boldsymbol{CE}\left(\boldsymbol{S}\left(D_{\boldsymbol{\theta}}\left(\mathbf{x}_0 + \mathbf{n}; \sigma, \mathcal{T}, \mathcal{M}, \mathbf{x}_{\mathcal{M}}\right) \odot \mathcal{M}\right), \mathcal{T}\right), \tag{9}$$

where $\boldsymbol{S}$ represents the STR function, which accepts an RGB image as input and outputs the recognition logits.

During the training process, the majority of the U-Net parameters are frozen to maintain the fundamental image generation capability of the original model conditioned by the visual context. Only the parameters of the cross-attention blocks are updated to learn a generalized visual representation of each character in the character set. We refer to this type of model fine-tuning as "knowledge complement". In this fine-tuning stage, the model attends to the character regions of the text images and encodes the character shape and appearance into "Keys" and "Values" of the cross-attention blocks. The complete objective of our training strategy can be expressed as a combination of the denoising score matching (DSM) loss, the local attention loss and the scene text recognition loss:

$$\mathcal{L} = \mathcal{L}_{DSM} + \lambda_{loc}\mathcal{L}_{loc} + \lambda_{str}\mathcal{L}_{str}. \tag{10}$$

3.3 Refinement of Noised Latent

Despite being constrained by the local attention loss, the fine-tuned model is still prone to producing spelling errors when rendering words in text images, such as missing some characters in a target word. We attribute this problem to a fundamental flaw in existing T2I models, i.e. catastrophic neglect. To address this issue, we implement noised latent refinement during the inference stage. Motivated by the generative semantic nursing approach introduced in [5], we design a new loss function $\mathcal{L}_{aae}$ with the aim of enhancing the maximum scores of the attention maps $\mathbf{A}_i^j$ corresponding to each character $\mathbf{c}^j$ within the region delineated by the binary mask $\mathcal{M}$:

$$\mathcal{L}_{aae}(\mathcal{A}, \mathcal{M}) = -\frac{1}{C}\sum_{i=1}^{C}\left\{\underset{1\le j\le N}{\boldsymbol{min}}\left(\boldsymbol{max}\left(\mathbb{G}\left(\mathbf{A}_i^j\right) \odot \mathcal{M}\right)\right)\right\}. \tag{11}$$

Our noised latent refinement process mainly consists of two steps: identifying an optimal initial noise $\mathbf{n}$, and optimizing the noised latent $\mathbf{z}_t$ at each timestep t. Initially, we sample Gaussian noise N times from the distribution $\mathcal{N}\left(\mathbf{0}, \sigma^2\boldsymbol{I}d\right)$. For each sampled noise, we swiftly execute the entire denoising process in a limited number (e.g., 2) of iterations and compute the corresponding objective $\mathcal{L}_{aae}$ at the final timestep. Subsequently, we select the noise with the minimum loss value as our initial noise $\mathbf{n}_{i^*}$. During the denoising process to get the final output, we refine the noised latent $\mathbf{z}_t$ using the gradient calculated based on the proposed objective $\mathcal{L}_{aae}$ at each timestep t:

$$\mathbf{z}_t' = \mathbf{z}_t - \alpha_t \cdot \nabla_{\mathbf{z}_t}\mathcal{L}_{aae}, \tag{12}$$

where α_t represents the learning rate used to update the noised latent $\mathbf{z}_t$ at each timestep t. The gradient $\nabla_{\mathbf{z}_t}\mathcal{L}_{aae}$ is computed in a backward manner through the parameters of the U-Net on the noised latent $\mathbf{z}_t$. The specifics of the refinement

process are outlined in Algorithm 1. We utilize the denoising algorithm proposed in [18]. Here, ***EulerStep*** denotes a single sampling step implemented using the Euler's method and $\boldsymbol{ODESchedule}(T)$ is the ODE scheduler which takes the number of ODE solver iterations T as input and outputs the σs of discretized sampling steps.

Algorithm 1 Denoising process with refinement

Input: A binary mask $\mathcal{M}$, a text condition $\mathcal{T}$, a masked image $\mathbf{x}_{\mathcal{M}} = (\boldsymbol{J} - \mathcal{M}) \odot \mathbf{x}_0$, a U-Net denoiser $D_{\boldsymbol{\theta}}$ and a latent decoder $\mathcal{D}$
Output: the denoised image $\hat{\mathbf{x}_0}$

$\{\sigma_2, \sigma_1\} \leftarrow \boldsymbol{ODESchedule}(2)$
for $i \leftarrow 1, 2 \dots N$ **do**
  $\mathbf{n}_i \sim \mathcal{N}\left(\mathbf{0}, \sigma_2^2 \boldsymbol{I}_d\right)$
  $\mathbf{d}, \mathcal{A}_2 \leftarrow D_{\boldsymbol{\theta}}(\mathbf{n}_i; \sigma_2, \mathcal{T}, \mathcal{M}, \mathbf{x}_{\mathcal{M}})$
  $\mathbf{z} \leftarrow \boldsymbol{EulerStep}(\mathbf{d}, \mathbf{n}_i, \sigma_2)$
  $\_, \mathcal{A}_1 \leftarrow D_{\boldsymbol{\theta}}(\mathbf{z}; \sigma_1, \mathcal{T}, \mathcal{M}, \mathbf{x}_{\mathcal{M}})$
  $\mathcal{L}_i \leftarrow \mathcal{L}_{aae}(\mathcal{A}_1, \mathcal{M})$
end for
$i^* \leftarrow \underset{1 \le i \le N}{argmin}\ \mathcal{L}_i$ ▷ select the best initial noise
$\mathbf{z}_T \leftarrow \mathbf{n}_{i^*}$
$\{\sigma_T, \sigma_{T-1}, \dots \sigma_1\} \leftarrow \boldsymbol{ODESchedule}(T)$
for $t \leftarrow T, T-1 \dots 1$ **do**
  $\_, \mathcal{A}_t \leftarrow D_{\boldsymbol{\theta}}(\mathbf{z}_t; \sigma_t, \mathcal{T}, \mathcal{M}, \mathbf{x}_{\mathcal{M}})$
  $\mathcal{L}_t \leftarrow \mathcal{L}_{aae}(\mathcal{A}_t, \mathcal{M})$
  $\mathbf{z}'_t \leftarrow \mathbf{z}_t - \alpha_t \cdot \nabla_{\mathbf{z}_t} \mathcal{L}_t$ ▷ refine the noised latent
  $\mathbf{d}_{t-1}, \_ \leftarrow D_{\boldsymbol{\theta}}(\mathbf{z}'_t; \sigma_t, \mathcal{T}, \mathcal{M}, \mathbf{x}_{\mathcal{M}})$
  $\mathbf{z}_{t-1} \leftarrow \boldsymbol{EulerStep}(\mathbf{d}_{t-1}, \mathbf{z}'_t, \sigma_t)$
end for
$\hat{\mathbf{x}_0} \leftarrow \mathcal{D}(\mathbf{z}_0)$
return $\hat{\mathbf{x}_0}$

4 Experiments

4.1 Datasets and Evaluation Metrics

To apply the training strategy mentioned in Sec. 3.2 and enhance the generalization capability of our proposed model, we require large-scale datasets, which should offer a diverse range of character samples, varying in shape and style. Ideally, the datasets should contain large numbers of text images, text annotations and the bounding boxes of text regions. Additionally, the character-level segmentation maps are also necessary. Considering these requirements, we have selected both synthetic and real-world datasets to constitute our training data:

– **SynthText in the Wild** [11] is a synthetically generated dataset, in which word instances are placed in natural scene images, while taking into account

the scene layout. The dataset consists of 800,000 images with approximately 8 million synthetic word instances. Each text instance is annotated with its text-string, word-level and character-level bounding-boxes, which we utilize to generate character-level segmentation maps.
- **LAION-OCR** [7] derives from the large-scale dataset LAION-400M [32]. It contains 9,194,613 filtered high-quality text images including advertisements, notes, posters, covers, memes, logos, etc. The authors of [7] trained a character-level segmentation model to obtain the segmentation maps of the text images.

For the purpose of validation, we gather datasets that include text images not previously encountered by the model. These datasets are derived from various tasks, encompassing scene text detection and segmentation.

- **ICDAR13** [17] is the standard benchmark for evaluating near-horizontal text detection, which contains 233 test images.
- **TextSeg** [37] is a multi-purpose text dataset focused on segmentation. It contains real-world text images collected from posters, greeting cards, covers, logos, road signs, billboards, digital designs, handwriting, etc. 340 images of them are for validation.
- **LAION-OCR** evaluation dataset. We partition a subset of the **LAION-OCR** dataset for the purpose of validation. The text images in this subset are not exposed to the model during the training phase.

We assess the performance of our proposed model in two aspects: image quality and text sequence accuracy. For the evaluation of image quality, we employ Fréchet Inception Distance (**FID**) [13] to measure the distance between the text images in the dataset and the images generated by our model and other baselines. Furthermore, we incorporate Learned Perceptual Image Patch Similarity (**LPIPS**) [43] as an additional metric to assess the quality of the generated images. The above metrics provide an indication of the visual coherence between the rendered text and its background. Given that our primary objective is to correct word spelling errors prevalent in existing diffusion models, we utilize an off-the-shelf scene text recognition (STR) model [2] to identify the rendered text. Subsequently, we employ sequence accuracy (**SeqAcc**) to evaluate the word-level correctness by comparing the STR result with the ground truth.

4.2 Implementation Details

UDiffText primarily comprises two components: a U-Net backbone and the proposed character-level text encoder. For the U-Net, we employ the pre-trained checkpoint of Stable Diffusion (v2.0) inpainting version. The model is fine-tuned using an image size of 512×512 on the SynthText dataset for 100k steps and then on the LAION-OCR dataset for an additional 100k steps. The training process utilizes a batch size of 64 and a learning rate of 5×10^{-5}. The U-Net encompasses 891M parameters, of which only 75.9M (the parameters of the cross-attention

blocks) are updated during training. As for the character-level text encoder, it undergoes initial training using the strategy outlined in Sec. 3.1 for 8k steps with a batchsize of 256 and a learning rate of 1×10^{-5}. Following this, it is frozen and connected to the U-Net for subsequent training. The proposed encoder comprises approximately 302M parameters. In the training stage, we set λ_{ce} to 0.1, λ_{loc} to 0.01 and λ_{str} to 0.001. During the inference stage, we employ 50 sampling steps and utilize a classifier-free guidance (CFG) scale of 5.0.

Fig. 4: Qualitative results on the scene/document/poster text inpainting task. The first column consists of the original images, while the second column comprises the input images with binary masks applied to the text region. The specific word to be generated is indicated at the left of each row.

4.3 Quantitative and Qualitative Results

To validate the superiority of our proposed method, we compare it with several scene text synthesis techniques, including the GAN-based method (MOSTEL [26]), and diffusion-based methods (DiffSTE [16] and TextDiffuser [7]). For better comparison, we evaluate all methods across two distinct tasks: scene text inpainting of the original text and the substitute text. In the case of the former, we employ the models to reconstruct the text image using the provided ground truth text label and binary mask. For the latter, we substitute the original text

in each image with a random word of equivalent length and evaluate the models by generating images containing the new text. The sequence accuracy (**SeqAcc**) for these tasks is denoted as **SeqAcc-O** and **SeqAcc-S**, respectively. We limit the text length in each instance to a maximum of 12 characters and randomly select 100 images from each dataset for testing.

The quantitative evaluation is conducted repeatedly 3 times for more convincing comparison and the mean value/standard deviation results are presented in Tab. 1. For TextDiffuser [7], we utilize their inpainting variant, where we render the desired text in a standard font (Arial) at the masked region as the input for their proposed segmentor. As for MOSTEL [26], we employ it to generate the text at the masked region and then integrate the output back into the original image. Their **FID** and **LPIPS** scores appear satisfactory, in part because the background remains unaltered. Furthermore, we also assess the performance of the pre-trained Stable Diffusion (v2.0) inpainting version as a baseline result. We set the prompt as "[word to be rendered]" for fair comparison. Overall, our method outperforms the baselines across all quantitative metrics, suggesting that our proposed model is capable of generating text images with superior sequence accuracy and quality, conditioned solely on the text label. For the qualitative results, we display the outputs of all aforementioned methods on the scene text inpainting task with substitute text. As illustrated in Fig. 4, our method yields the most visually pleasing results, characterized by high text rendering accuracy and visual context coherency. For more qualitative results, please refer to Sec. B of our supplementary material.

Table 1: Quantitative comparison between our method and four baselines. **ICDAR13 (8ch)** denotes that we restrict the text length to be no more than 8 characters for the purpose of evaluating short word rendering performance. The items in the table contain the mean/standard of the data points and the best scores are highlighted in bold.

Method	SeqAcc-O(%)↑				SeqAcc-S (%)↑				FID↓	LPIPS↓
	ICDAR13 (8ch)	ICDAR13	TextSeg	LAION-OCR	ICDAR13 (8ch)	ICDAR13	TextSeg	LAION-OCR		
MOSTEL	74.7/2.87	67.3/2.50	64.3/2.87	70.0/2.16	34.3/2.49	28.0/2.45	24.7/2.05	48.3/2.62	20.8/1.25	0.0634/0.0022
SD-Inpainting	33.0/1.41	28.3/0.94	12.3/1.25	15.0/0.82	7.0/0.82	6.7/0.47	4.3/0.47	6.0/0.81	26.6/1.43	0.0697/0.0018
DiffSTE	44.3/2.50	37.3/2.87	49.3/3.30	40.3/2.49	34.7/3.30	29.0/3.27	46.7/2.87	36.0/2.94	52.8/1.34	0.1069/0.0065
TextDiffuser	85.7/1.25	80.7/1.25	68.0/1.63	79.7/2.05	81.3/1.70	74.3/2.49	65.7/2.05	71.3/1.70	34.1/1.79	0.0881/0.0049
Ours	**93.3/1.70**	**91.0/1.63**	**92.3/1.25**	**90.3/1.25**	**84.0/1.63**	**82.7/1.25**	**84.3/1.25**	**79.0/1.63**	**19.7/1.89**	**0.0574/0.0011**

4.4 Ablation Study

To assess the efficacy of each design choice in our method, we perform an ablation study on various settings, which include: (1) Base: The inpainting version of the pre-trained Stable Diffusion (v2.0), which uses the CLIP text encoder to obtain conditional embeddings. (2) CL Encoder: We employ our proposed character-level text encoder (CL Encoder) as a replacement for the CLIP encoder. (3) L_{loc}: We incorporate the proposed local attention loss into the basic diffusion loss to serve as the training objective. (4) L_{str}: We introduce the scene text recognition loss for additional supervision. (5) Refinement: We apply the refinement of noised

Table 2: Ablation study results.

(a) Ablation study results on different design choices of the proposed method.

Setting	**SeqAcc-O (%)↑**
Base	8.0
+ CL encoder	40.0
+ L_{loc}	54.0
+ L_{str}	65.0
+ Refinement	**76.0**

(b) Ablation study results on the text encoder trained with different objectives.

Setting	ByT5	Ours-CLIP loss	Ours-Full loss
SeqAcc-S (%) ↑	10.0	17.0	**62.0**

(c) Ablation study results on the hyper-parameter λ_{loc} and λ_{str}.

λ_{loc}	0.1	0.01	0.001
SeqAcc-S (%) ↑	5.0	62.0	41.0

λ_{str}	0.1	0.01	0.001
SeqAcc-S (%) ↑	8.0	18.0	62.0

latent, as mentioned in Sec. 3.3, at the inference stage to enhance text accuracy. We train the model under all the above settings on the SynthText dataset for 6k steps and test them on the corresponding evaluation set. The quantitative results of sequence accuracy (**SeqAcc-O**) are presented in Tab. 2 (a), which indicate that the whole model outperforms the other variants.

In our work, we utilize a relatively light-weight CL text encoder instead of the ByT5 text encoder to avoid unacceptable computational waste. Our motivation of introducing the CE loss is to supervise the CL text encoder to learn a codebook with more divisible embeddings. To prove the above assumption, we train the whole model three times using ByT5 text encoder and our CL text encoder with the classic CLIP loss and the proposed full loss, respectively. As shown in Tab. 2 (b), the results prove that our CL text encoder helps to achieve much better performance compared to ByT5-base with the similar number of params and the CE loss helps to achieve much better performance. We also conduct ablation study on the hyper-parameter λ_{loc} and λ_{str} to indicate the influence of different weights of the loss components, as shown in Tab. 2 (c).

To further illustrate the efficacy of our character-level text encoder and local attention loss, we compare the performance of our UDiffText with that of Stable Diffusion. In a specific generation scenario, we extract the attention maps from the U-Net model during an intermediate inference step. As depicted in Fig. 7 of our supplementary material, it is evident that our UDiffText focuses on the precise regions of each rendered character, whereas Stable Diffusion exhibits ambiguous attention areas within the rendered word, leading to incorrect results and attention maps devoid of meaningful information. This experiment indicates that the local attention loss indeed imposes an effective constraint on the attention maps, thereby enhancing the interpretability of our proposed method.

4.5 Applications

Scene text inpainting. Taking an arbitrary image, a binary mask and a text sequence as input, UDiffText generates a modified image with the desired text rendered in a specific region defined by the mask. This inpainting-based archi-

tecture makes the proposed method suitable for a variety of inpainting-like text rendering applications. As demonstrated in Fig. 1 (a)(b) and Fig. 4, our method can be applied to tasks involving the synthesis of scene text in real-world images and scanned documents. Additionally, the proposed UDiffText has the potential to be applied to construct large-scale scene text image datasets, given its capability to generate context-coherent text images that do not exist in the real world. Moreover, our UDiffText can also be applied to graphic design tasks like poster design and advertisement design.

T2I generation with accurate text content. Leveraging the text inpainting capability of our proposed model, we devise a two-stage method for T2I generation that ensures accurate text rendering, as shown in Fig. 1 (c). Specifically, in our experiments, we first utilize the large-scale T2I model [3, 25] to produce a preliminary result using the prompt template generated by a LLM (e.g., GPT3.5). Then, we employ a previous SOTA text spotting model [41] to mask the text region in the generated image. At last, our UDiffText is applied to the masked image to produce the final output, which features accurate text and a consistent style (see Sec. C of our supplemental material). It should be noted that we drop and regenerate the image when no text is detected. Depending on the excellent performance of the spotting model, ill-fitting text bounding-boxes are barely encountered in our experiments. Furthermore, we also quantitatively evaluate our method using the SimpleBench prompt templates proposed in [40]. Experimental results show that our approach significantly improve the average text rendering accuracy of the pre-trained SDXL model from 8.0% to 60.0%.

5 Limitations

Though simple and effective, the propose method still has some limitations such as limited number of characters in generated images and lack of ability to preserve the style of the original text. Please refer to Sec. G of the supplemental material for a more detailed discussion and some failure cases.

6 Conclusion

In this paper, we proposed UDiffText, a novel method for high-quality text synthesis in arbitrary images using character-aware diffusion models. We designed and trained a character-level text encoder that provides robust text embeddings and fine-tuned the diffusion model with local attention control and scene text recognition supervision. Our method can generate coherent images with accurate text and can be used for arbitrary text generation, scene text inpainting and T2I generation with precise text content. We demonstrated the effectiveness of our method through extensive experiments and comparisons with existing methods, showing the superiority of the proposed UDiffText to the state of the art in terms of both text rendering accuracy and visual context coherency. In the future, we plan to explore more ways to improve the controllability and diversity of our method, and extend it to other text-related image synthesis tasks.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant No.: 62372015), Center For Chinese Font Design and Research, Key Laboratory of Intelligent Press Media Technology, and State Key Laboratory of General Artificial Intelligence.

References

1. Atienza, R.: Vision transformer for fast and efficient scene text recognition. In: Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part I 16. pp. 319–334. Springer (2021)
2. Bautista, D., Atienza, R.: Scene text recognition with permuted autoregressive sequence models. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII. pp. 178–196. Springer (2022)
3. 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`
4. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18392–18402 (2023)
5. Chefer, H., Alaluf, Y., Vinker, Y., Wolf, L., Cohen-Or, D.: Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. ACM Transactions on Graphics (TOG) **42**(4), 1–10 (2023)
6. Chen, H., Xu, Z., Gu, Z., Li, Y., Meng, C., Zhu, H., Wang, W., et al.: Diffute: Universal text editing diffusion model. Advances in Neural Information Processing Systems **36** (2024)
7. 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)
8. Feng, W., He, X., Fu, T.J., Jampani, V., Akula, A., Narayana, P., Basu, S., Wang, X.E., Wang, W.Y.: Training-free structured diffusion guidance for compositional text-to-image synthesis. arXiv preprint arXiv:2212.05032 (2022)
9. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A.H., Chechik, G., Cohen-Or, D.: An image is worth one word: Personalizing text-to-image generation using textual inversion. arXiv preprint arXiv:2208.01618 (2022)
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
11. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
12. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022)

13. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
14. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems **33**, 6840–6851 (2020)
15. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
16. 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)
17. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L.G., Mestre, S.R., Mas, J., Mota, D.F., Almazan, J.A., De Las Heras, L.P.: Icdar 2013 robust reading competition. In: 2013 12th international conference on document analysis and recognition. pp. 1484–1493. IEEE (2013)
18. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. Advances in Neural Information Processing Systems **35**, 26565–26577 (2022)
19. Krishnan, P., Kovvuri, R., Pang, G., Vassilev, B., Hassner, T.: Textstylebrush: Transfer of text aesthetics from a single example. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)
20. Kumari, N., Zhang, B., Zhang, R., Shechtman, E., Zhu, J.Y.: Multi-concept customization of text-to-image diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1931–1941 (2023)
21. 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)
22. Ma, J., Zhao, M., Chen, C., Wang, R., Niu, D., Lu, H., Lin, X.: Glyphdraw: Learning to draw chinese characters in image synthesis models coherently. arXiv preprint arXiv:2303.17870 (2023)
23. Mou, C., Wang, X., Xie, L., Wu, Y., Zhang, J., Qi, Z., Shan, Y.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 4296–4304 (2024)
24. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021)
25. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952 (2023)
26. Qu, Y., Tan, Q., Xie, H., Xu, J., Wang, Y., Zhang, Y.: Exploring stroke-level modifications for scene text editing. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 2119–2127 (2023)
27. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022)
28. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., 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)

29. Roy, P., Bhattacharya, S., Ghosh, S., 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)
30. Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., Norouzi, M.: Palette: Image-to-image diffusion models. In: ACM SIGGRAPH 2022 Conference Proceedings. pp. 1–10 (2022)
31. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in Neural Information Processing Systems **35**, 36479–36494 (2022)
32. Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., Komatsuzaki, A.: Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. arXiv preprint arXiv:2111.02114 (2021)
33. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
34. Tewel, Y., Gal, R., Chechik, G., Atzmon, Y.: Key-locked rank one editing for text-to-image personalization. In: ACM SIGGRAPH 2023 Conference Proceedings. pp. 1–11 (2023)
35. Wu, L., Zhang, C., Liu, J., Han, J., Liu, J., Ding, E., Bai, X.: Editing text in the wild. In: Proceedings of the 27th ACM international conference on multimedia. pp. 1500–1508 (2019)
36. Xiao, G., Yin, T., Freeman, W.T., Durand, F., Han, S.: Fastcomposer: Tuning-free multi-subject image generation with localized attention. arXiv preprint arXiv:2305.10431 (2023)
37. Xu, X., Zhang, Z., Wang, Z., Price, B., Wang, Z., Shi, H.: Rethinking text segmentation: A novel dataset and a text-specific refinement approach. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12045–12055 (2021)
38. Xue, L., Barua, A., Constant, N., Al-Rfou, R., Narang, S., Kale, M., Roberts, A., Raffel, C.: Byt5: Towards a token-free future with pre-trained byte-to-byte models. Transactions of the Association for Computational Linguistics **10**, 291–306 (2022)
39. Yang, Q., Huang, J., 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)
40. 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)
41. Ye, M., Zhang, J., Zhao, S., Liu, J., Liu, T., Du, B., Tao, D.: Deepsolo: Let transformer decoder with explicit points solo for text spotting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19348–19357 (2023)
42. Zhang, L., Rao, A., 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)
43. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., 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)