EasyText: Controllable Diffusion Transformer for Multilingual Text Rendering

Runnan Lu[1] **Yuxuan Zhang**[2] **Jiaming Liu**[3] **Haofan Wang**[4] **Yiren Song**[1*]

[1]National University of Singapore
[2]The Chinese University of Hong Kong
[3]Alibaba
[4]Liblib AI
songyiren725@gmail.com

Abstract

Generating accurate multilingual text with diffusion models has long been desired but remains challenging. Recent methods have made progress in rendering text in a single language, but rendering arbitrary languages is still an under-explored area. This paper introduces EasyText, a text rendering framework based on DiT (Diffusion Transformer), which connects denoising latents with multilingual character tokens encoded as character tokens. We propose character positioning encoding and position encoding interpolation techniques to achieve controllable and precise text rendering. Additionally, we construct a large-scale synthetic text image dataset with 1 million multilingual image-text annotations as well as a high-quality dataset of 20K annotated images, which are used for pretraining and fine-tuning respectively. Extensive experiments and evaluations demonstrate the effectiveness and advancement of our approach in multilingual text rendering, visual quality, and layout-aware text integration.

Code — https://github.com/songyiren725/EasyText
Datasets —
https://huggingface.co/datasets/lllrrnn/EasyText
Extended version — https://arxiv.org/abs/2505.24417

1 Introduction

Scene text rendering is crucial for various real-world applications. However, most existing methods such as TextDiffuser (Chen et al. 2023a, 2024), Diff-font (He et al. 2024) and modern commercial models like FlUX-dev (Labs 2024) and Ideogram (Ideogram Inc. 2025), are primarily limited to English, making multilingual text rendering still a challenging task. Glyph-ByT5-V2 (Liu et al. 2024b) was one of the earliest and most representative works to enable multilingual text rendering by introducing a specially designed glyph encoder and employing a multi-stage training strategy.

Inspired by how humans learn to write, we derive several key insights: (1) Imitative writing is considerably easier than recalling—humans typically begin by mimicking before advancing to memory-based writing. (2) Once familiar with one language, humans naturally develop the ability to reproduce text in other unfamiliar languages even without understanding them—treating it more like drawing than writing. Motivated by this, we argue that training AI to "imitate" rather than "recall" is a more efficient and effective strategy for text rendering.

The task of scene text rendering faces several key challenges: (1) Multilingual character modeling is highly complex—for example, Chinese alone has over 30,000 characters. Extending to multilingual settings drastically expands the character space, making joint modeling harder, especially for rare or low-frequency characters. This is further complicated by language imbalance and font variability. (2) Text-background integration is often unnatural; existing methods struggle to blend rendered text with scene content, leading to visual artifacts such as disconnection or pasted-on effects that undermine image realism. (3) Preserving generative priors is difficult, as fine-tuning on large-scale text-image datasets improves rendering capabilities but often degrades the model's general image generation ability.

To this end, we present EasyText (shown in Fig. 1), a multilingual text image generation framework based on Diffusion Transformers (Peebles and Xie 2023; Vaswani et al. 2017). We encode text into font tokens via a VAE and concatenate them in the latent space with denoised latents. Leveraging the in-context capabilities of DiT, EasyText achieves high-quality and accurate text rendering. Additionally, we propose a simple yet effective position control strategy called Implicit Character Position Alignment, which allows for precise control of character positions through positional encoding interpolation and replacement—enabling both position-aware rendering and layout-free generation.

EasyText is also highly data-efficient. Unlike Glyph-ByT5-V1/V2 (Liu et al. 2024a,b), which rely on contrastive synthetic data or massive real-world text datasets, our method simply overlays text randomly on natural images during the pretraining stage to learn glyph features. To encourage the model to learn glyph imitation rather than simple shape copying, we employ a multi-font mapping approach. Multiple different fonts are overlaid in the synthetic training images, while the condition image uses only a standard font. Afterwards, we fine-tune a lightweight LoRA (Hu et al. 2022) on only 20K high-quality multilingual scene text images, enhancing the visual consistency between text and background. As shown in Table 1, we highlight some

*Corresponding Author.

Functionality	Position Control	Irregular Regions	Multi-Lingual	Long Text Rendering	Text-Image Blending	Unseen Characters
Glyph-SDXL-v2	✓	✗	✓	✓	✗	✗
AnyText	✓	✓	✓	✗	✓	✗
SD3.5	✗	✗	✗	✗	✓	✗
FLUX-dev	✗	✗	✗	✗	✓	✗
Jimeng AI 2.1	✗	✗	✓	✓	✓	✗
EasyText	✓	✓	✓	✓	✓	✓

Table 1: Functionality evaluation of EasyText in comparison to other competitors.

of the capabilities demonstrated by EasyText including: (a) **Position control**, precisely positioning text in specific locations within an image; (b) **Irregular regions**, enabling it to handle text rendering in irregular regions such as slanted or curved regions; (c) **Multilingual text handling**, supporting text generation in multiple languages such as Chinese, English, Japanese, etc; (d) **Long text rendering**, which can render extended text passages; (e) **Text-image blending**, which integrates text seamlessly into images, maintaining natural visual consistency; and (f) **Unseen character generalization**, allowing the model to exhibit generalization capability on unfamiliar and unseen characters.

In summary, our contributions are:

1. We propose EasyText, a framework that teaches AI to "imitate" rather than "recall", achieving high-quality multilingual text rendering by harnessing the in-context learning power of Diffusion Transformers.

2. We introduce Implicit Character Position Alignment, which precisely controls text placement via position encoding operations and supports layout-free generation.

3. Extensive experiments demonstrate the effectiveness and simplicity of our method, showing superior performance on challenging scenarios such as long text, multi-text layouts, irregular regions, and unseen characters.

2 Related Works

2.1 Diffusion Models

In recent years, diffusion models (Liang et al. 2024; Song, Meng, and Ermon 2020) have achieved remarkable progress and emerged as a widely used approach for image generation, offering strong capabilities in producing high-quality and diverse visual content through iterative denoising. Their strong capacity for modeling complex data distributions has enabled broad applicability, ranging from image synthesis and editing (Brooks, Holynski, and Efros 2023; Hertz et al. 2022) to video generation (Bar-Tal et al. 2024). Notable examples include Stable Diffusion (Rombach et al. 2022), its enhanced version SDXL (Podell et al. 2023), and subsequent variants, which have demonstrated the scalability and effectiveness of text-to-image diffusion models for high-quality image synthesis. Recently, the use of transformer-based denoisers—particularly the Diffusion Transformer (DiT) (Peebles and Xie 2023; Feng and Zhang 2023; Feng et al. 2025; Feng and Zhang 2024), which replaces U-Net with a transformer backbone—has gained significant traction, and has been adopted in many state-of-the-art models such as FlUX-dev (Labs 2024), Stable Diffusion 3.5 (Esser et al. 2024), and PixArt (Chen et al. 2023b).

2.2 Condition-guided Diffusion Models

Condition-guided diffusion models incorporate external signals—such as spatial structure or semantic reference—into the generative process, enabling control over layout (Li et al. 2024b; Zhang et al. 2025), content (Zhang et al. 2024b,c; Wang et al. 2024b; Zhang et al. 2024d,e; Song et al. 2024), motion (Ma et al. 2025e, 2023, 2024, 2025d,b,c), and identity (Wang et al. 2024a). These methods facilitate consistent synthesis, alignment with user intent, and support for flexible customization. Early implementations based on U-Net architectures adopted two main paradigms: attention-based conditioning, which integrates semantic features via auxiliary encoders, and residual-based fusion, which injects spatial features into intermediate layers. ControlNet (Zhang, Rao, and Agrawala 2023) and T2I-Adapter (Mou et al. 2024) are representative examples that improve spatial consistency and layout control under this framework. With the adoption of transformer-based diffusion models, recent approaches reformulate both semantic and spatial conditions as token sequences and integrate them into the generation process through multi-modal attention or token concatenation mechanisms, as demonstrated in DiT-based systems like Omini-Control (Tan et al. 2024) . This unified formulation advances the development and application of conditional image generation (Wan et al. 2024; Song, Liu, and Shou 2025; Song, Chen, and Shou 2025; Huang et al. 2025; Guo et al. 2025), improving scalability, simplifying model design, and enabling efficient handling of multiple conditions.

2.3 Visual Text Generation

Text generation and rendering is a classic task, where early methods mainly relied on generative adversarial networks (GANs) (Wu et al. 2019; Cha et al. 2020; Azadi et al. 2018; Tang et al. 2022) and vector stroke techniques (Thamizharasan et al. 2024; Song et al. 2023; Song and Zhang 2022). Recent work on visual text generation with diffusion models primarily focuses on optimizing text encoding and spatial control mechanisms to improve the fidelity and controllability of rendered text. Character-aware encoders are increasingly adopted: GlyphDraw (Ma et al. 2023), GlyphControl (Yang et al. 2023), and AnyText (Tuo

Figure 1: Text-rendered results generated by EasyText, which supports text rendering in over ten languages and produces high-quality results. It can render text either with explicit positional control or in a layout-free manner, and effectively handles curved and slanted regions. The additional texts shown below each prompt denote the target texts to be rendered in the image which are not part of the input prompt.

et al. 2023) embed glyph or OCR features into conditional inputs, while TextDiffuser-2 (Chen et al. 2024) leverages character-level tokenization to improve alignment. However, semantic bias and tokenizer limitations still persist. Spatial control has been addressed by introducing explicit layout-related conditions. TextDiffuser (Chen et al. 2023a) and ControlText (Jiang et al. 2025) use segmentation or layout masks, and UDiffText (Zhao and Lian 2023) and Brush Your Text (Zhang et al. 2024a) refine attention to enforce region-level alignment. For multilingual rendering, works like AnyText (Tuo et al. 2023), GlyphControl (Yang et al. 2023), and Glyph-ByT5-v2 (Liu et al. 2024b) bypass tokenizers using glyph encoders or tokenizer-free models. Yet, rendering quality remains limited by encoder–diffusion compatibility. Newer models (GlyphDraw2 (Ma et al. 2025a), JoyType (Li et al. 2024a), AnyText2 (Tuo, Geng, and Bo 2024), FonTS (Shi et al. 2024), RepText (Wang et al. 2025)) improve layout precision, while closed models (Kolors 2.0 (Kolors Team 2024), WordCon (Shi et al. 2025), Seedream 3.0 (Gao et al. 2025), GPT-4o (OpenAI 2024)) show promising results but often suffer from limited spatial controllability and suboptimal performance in multi-text layouts. Despite progress, challenges remain in complex layouts and multilingual text, motivating further integration of glyph, position, and semantic signals.

3 Method

The training of our method is based on the open-source base model FLUX (refer to Appendix Section A for details). Section 3.1 outlines the overall architecture; Section 3.2 describes the target text condition representation; Section 3.3 details the Implicit Character Position Alignment; and Section 3.4 explains the paired dataset construction.

3.1 Overall Architecture

The training pipeline adopts a two-stage strategy. The first stage involves training on a large-scale, synthetically generated dataset with balanced multilingual coverage, enabling the model to learn glyph generation across diverse scripts and accurate spatial mapping. In the second stage, we fine-tune the model on a high-quality annotated dataset to improve both the aesthetic quality of rendered text and its integration with complex scene content. For conditioning, the input condition image (I_c) is encoded into the same latent space as the target image (I_{tar}) via a VAE. Its positional encoding is aligned with the target region through Implicit Character Position Alignment, then concatenated with the target image's position encoding before being passed to the FLUX DiT block. (Shown in Fig. 2) Image-conditioned LoRA adaptation is then applied, with both the VAE and text encoder kept frozen.

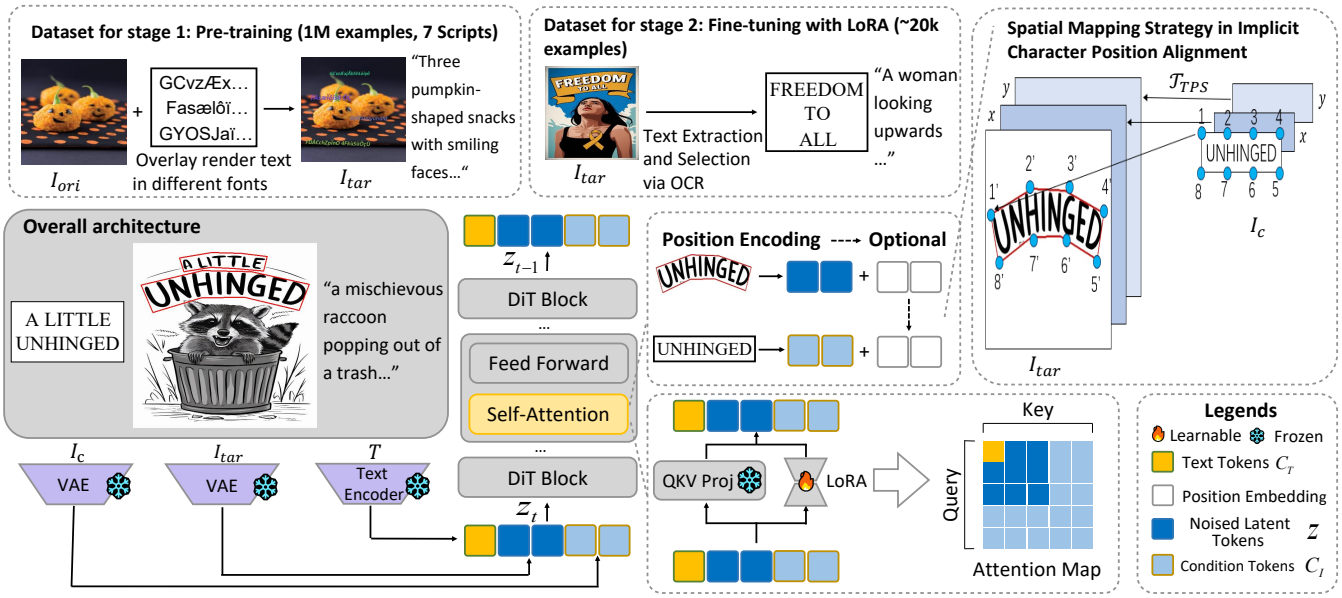

Figure 2: Overview of EasyText. A two-stage training strategy is used: large-scale pretraining for glyph generation and spatial mapping, followed by fine-tuning for visual-text integration and aesthetic refinement. Character positions from the condition input are implicitly aligned with target regions, and training proceeds with image-conditioned LoRA.

3.2 Character Representation in EasyFont

In contrast to conventional font conditioning approaches that typically employ symbolic or parametric representations, our methodology innovatively adopts a visually-grounded paradigm inspired by glyph morphology. The key differentiation of our character representation lies in its image-based conditioning architecture, where discrete image patches serve as fundamental conditioning units. This design philosophy stems from two critical observations:

First, we establish a unified multilingual representation to address the intrinsic typographic differences between writing systems. For alphabetic scripts (e.g., English), we use 64-pixel-high images with widths adaptive to text length, naturally capturing the connected structure of alphabetic word formations. This horizontal layout preserves crucial inter-character relationships and spacing conventions unique to Western typography. Second, for logographic systems (Chinese, Japanese, etc.), we assign a fixed-size square image to each character. For Chinese characters in particular, the image size is set to 64×64 pixels. This configuration respects the isolated nature of ideographic characters while maintaining consistent resolution. The square aspect ratio optimally captures the balanced structure inherent to characters, where stroke complexity is uniformly distributed within a square design space.

Our condition image efficiently captures the rich visual features of glyphs while providing a significantly more compact representation than many existing methods that use layout inputs matching the target image resolution. By including only the text to be rendered, it typically requires less than one-tenth of the spatial size, substantially reducing computational overhead.

3.3 Implicit Character Position Alignment

To enable flexible and precise spatial control over the rendered text within the condition image, we introduce an *Implicit Character Position Alignment* (ICPA) mechanism, which maps the spatial coordinates of the rendered text in the target image (even in irregular regions) onto the corresponding characters in the condition image. Given a conditional image patch $\mathbf{P}_c \in \mathbb{R}^{64\times W_c\times 3}$ containing source typography features—where W_c denotes the total width of the conditional image patch, and a target rendering region Ω_t in the output image, our method establishes position-aware feature correspondence through positional encoding extrapolation.

Linear Alignment via Affine Transform. Linearly interpolate the position in the condition image to the position in the target bounding box (an axis-aligned rectangular region in this case) $\Omega_t = [x_1 : x_2, y_1 : y_2]$. Assuming $u \in [0, W_c - 1]$ and $v \in [0, 63]$ (i.e. the condition image spans indices 0 to W_c horizontally and 0 to 63 vertically), the affine mapping $\mathcal{T}_{\text{aff}} : (u, v) \mapsto (x, y)$ is defined as:

$$\mathcal{T}_{\text{aff}}(u,v) = \Big(x_1 + \frac{u}{W_c - 1}(x_2 - x_1), \\ y_1 + \frac{v}{63}(y_2 - y_1)\Big), \tag{1}$$

which maps the normalized horizontal coordinate $u/(W_c - 1)$ to a point between x_1 and x_2, and similarly $v/63$ to a point between y_1 and y_2, achieving a linear scaling and translation of the coordinates in condition image into the target domain. The positional encoding in condition image is then re-aligned by applying the affine transform, yielding a transformed position $(u', v') = (x, y) = \mathcal{T}_{\text{aff}}(u, v)$. These transformed coordinates are used to update the positional encoding in condition image, enabling spatial consistency

between the condition and target images.

Nonlinear Alignment via Thin-Plate Spline Interpolation. For nonlinear spatial alignment, we adopt Thin-Plate Spline (TPS) interpolation, which defines a smooth mapping that exactly fits a set of control point correspondences between the source patch and the target region, while minimizing second-order bending distortion. Here, Ω_t denotes an irregular region. We obtain K control points along its boundary to form a deformation region that closely fits and fully covers the rendered text. These points act as spatial anchors for the TPS transformation. Let $\{(u_i, v_i)\}_{i=1}^{K}$ be K landmark points in the condition image P_c and $\{(x_i, y_i)\}_{i=1}^{K}$ their corresponding positions in the target region Ω_t. The TPS mapping $\mathcal{T}_{\text{TPS}} : (u, v) \mapsto (x, y)$ can be expressed as an affine base plus a radial basis deformation term:

$$\mathcal{T}_{\text{TPS}}(u, v) = A \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} + \sum_{i=1}^{K} w_i \, \phi\Big(\sqrt{(u - u_i)^2 + (v - v_i)^2}\Big), \quad \phi(r) = r^2 \log r, \tag{2}$$

where $A \in \mathbb{R}^{2\times 3}$ represents the affine part of the transformation and $w_i \in \mathbb{R}^2$ are the TPS warp coefficients associated with each control point. The kernel $\phi(r) = r^2 \ln r$ is the fundamental solution of the biharmonic equation in 2D. The parameters A and w_i are determined by solving the interpolation constraints

$$\mathcal{T}_{\text{TPS}}(u_i, v_i) = (x_i, y_i), i = 1, \ldots, K, \tag{3}$$

together with additional conditions to ensure a well-posed solution ($\sum_{i=1}^{K} w_i = 0$, $\sum_{i=1}^{K} w_i u_i = 0$, $\sum_{i=1}^{K} w_i v_i = 0$). Similar to linear alignment, transformed position $(u', v') = (x, y) = \mathcal{T}_{\text{TPS}}(u, v)$ are used to update the positional encoding in condition image.

Layout-Free Position Alignment via Positional Offset Injection. To enable flexible layout-free rendering, we also introduce an effective positional offset strategy. We shift the positional encoding of the conditional image by a fixed scalar offset w_t, which represents the width of the target image. Let (u, v) index the spatial position in $\mathbf{P}_c$. Updated coordinates (u', v') are obtained by applying a positional offset to the original coordinates:

$$\begin{cases} u' = u + w_t \\ v' = v \end{cases}. \tag{4}$$

This ensures positional encoding uniqueness without binding the conditional image to any specific target location, enabling more flexible text rendering.

3.4 EasyText Dataset Construction

We construct two datasets tailored to the specific needs of the pretraining and fine-tuning stages.

Large-Scale Synthetic Dataset. This dataset is designed to provide broad coverage across scripts and languages. Target images are generated by rendering multilingual text onto background images using script-based synthesis. The dataset includes scripts such as Latin, Chinese, Korean, and others, resulting in nearly 1 million samples. To reduce redundancy and improve efficiency, languages sharing the same writing system are grouped by script—for example, English, Italian, and German are treated as a single Latin script class. The rendered text is composed of random combinations of characters from the corresponding script, covering its full character set. To encourage the model to learn generalizable glyph representations rather than simply copying shapes from the condition image, the target text is rendered using diverse fonts, while the condition image is rendered using a standardized font. This font decoupling enables robust learning of character structure across styles. Dataset statistics are summarized in Table 2.

Script Type	Large-Scale Pre-Training			Fine-Tuning
	Unique Chars	Font Types	Samples	Samples
Chinese	7000	18	230K	5.5K
Korean	4308	13	120K	0.1K
Japanese	2922	17	120K	0.1K
Thai	2128	3	120K	✗
Vietnamese	91	19	120K	✗
Latin	104	30	180K	15K
Greek	79	19	120K	✗

Table 2: Statistics of multilingual data used in large-scale pre-training and fine-tuning.

High-Quality Human-Annotated Dataset. The second dataset consists of approximately 20k high-quality image–text box pairs, primarily in Chinese and English. Text regions are extracted using PP-OCR (Du et al. 2020) and filtered to ensure annotation quality. In these examples, text is more naturally integrated into scene content, often with stylized typography and visually consistent stroke patterns. The rendered text is not included explicitly in the prompt, but instead triggered by placeholder tokens (e.g., <sks1>, <sks2>), supporting semantic alignment between text and background. This dataset facilitates better visual-text fusion and improves typographic aesthetics.

This two-stage training strategy reduces the need for large-scale text-image datasets, requiring only a small fine-tuning set—particularly well-suited in multilingual settings where annotated data is scarce.

4 Experiments

4.1 Experimental Settings

Implementation Details. Our method is implemented based on the open-source FLUX-dev framework with LoRA-based parameter-efficient tuning. In the first stage, we set the LoRA rank to 128 and train on 4 H20 GPUs. The model is first trained at 512×512 resolution for 48,000 steps with a batch size of 24, and then at 1024×1024 for 12,000 steps. In the second stage, we reduce the LoRA rank to 32 and fine-tune at 1024 resolution for 10,000 steps with a batch size of 8. We use the AdamW optimizer with a learning rate of 1e-4.

Evaluation Metric. To evaluate the accuracy of multilingual text generation, we adopt two complementary metrics: (i) Character-level precision, which measures the correctness of generated characters relative to ground-truth texts. This reflects the model's ability to produce accurate predictions at the most basic unit level for each language. (ii) Sentence-level precision, which assesses whether the entire

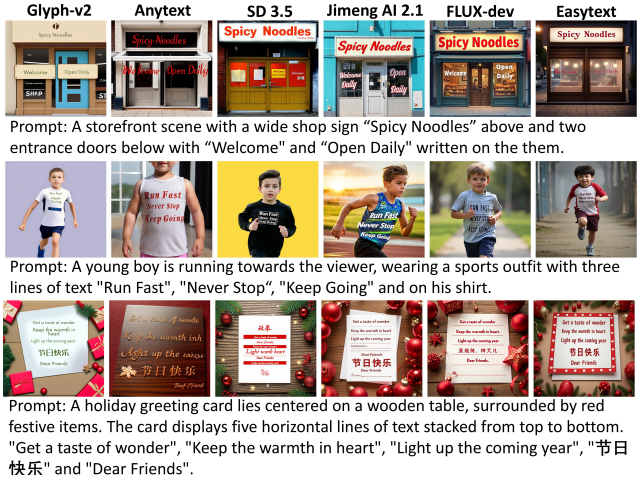

Figure 3: Qualitative comparison of EasyText with other methods, focusing on the generation quality of both text and images, reveals that EasyText demonstrates outstanding performance.

Figure 4: Qualitative comparison of EasyText across multiple languages of the same prompt.

text line content within each controlled text box is rendered completely correctly. This captures the overall consistency and correctness at the region level.

Multilingual Benchmark. We construct a multilingual benchmark with 90 language-agnostic prompts. For each language, prompts are paired with language-specific text to generate condition images while preserving semantic intent. Four images are generated per prompt-language pair. For logographic scripts, the average number of characters per text box is 7.1, and for alphabetic scripts, it is 14.3. Each image contains an average of 1.7 text boxes. To evaluate the generated results, three volunteers were tasked with assessing the accuracy of the text in the images, ensuring reliable multilingual generation quality.

Method	English		Chinese	
	Chars Pre	Sen Pre	Chars Pre	Sen Pre
EasyText (pretrain)	**0.9968**	**0.9813**	**0.9344**	0.6562
EasyText (fine-tune)	0.9945	0.9625	0.9312	0.6438
Glyph-SDXL-v2	0.9959	0.9688	0.9258	0.6250
FLUX	0.9180	0.7062	✗	✗
Stable Diffusion 3.5	0.9556	0.7938	✗	✗
Jimeng 2.1	0.9812	0.8687	0.9214	**0.6813**
ANYTEXT	0.8978	0.6364	0.8824	0.6071

Table 3: Character/Sentence level precision for English and Chinese across different methods.

4.2 Comparison Results

We evaluate our model on a multilingual benchmark against state-of-the-art commercial models (e.g., FLUX (Labs 2024), Jimeng AI (ByteDance 2024), SD3.5 (Esser et al. 2024)) and other methods (e.g., Glyph-ByT5-v2 (Liu et al. 2024b), AnyText (Tuo et al. 2023)), following the benchmark and metrics outlined in Sec. 4.1. For languages with limited support in prior rendering methods, we compare primarily with Glyph-SDXL-v2. All baseline models are evaluated using their official inference settings. For each sample, four images are generated with identical image descriptions and target texts across all methods. For models lacking conditional input support, the target text is included in the prompt; evaluation focuses solely on text accuracy, regardless of spatial layout or surrounding content.

Comprehensive Quality Assessment. Beyond precision evaluation, we assess generation quality using CLIPScore and OCR accuracy for objective comparison against existing baselines. We further incorporate GPT-4o assessment and a user study across four subjective criteria: Image Aesthetics, Text Aesthetics, Text Quality, and Text-Image Fusing, to evaluate overall fidelity and alignment.

The results show that our model surpasses competing methods in text rendering precision across several languages, including English and Italian (Table 4) and achieves strong character-level accuracy in Chinese (Table 3), though slightly behind Jimeng AI at the sentence level. It also performs well in unsupported languages like Thai and Greek. In Table 5, it also leads in OCR accuracy and improves over FLUX in CLIPScore, indicating higher visual-text alignment. These trends are further supported by GPT-4o evaluation results. After pretraining, the model demonstrates strong text rendering performance from condition images. However, it shows limited text-image coherence, as indicated by lower CLIPScore and GPT-4o evaluations. Fine-tuning alleviates this issue, significantly improving CLIPScore, Text Aesthetics, and overall visual-text alignment.

4.3 Qualitative Results

We provide qualitative comparisons of multilingual text rendering, including Chinese, English, and other languages. Results are shown in Fig. 4. Compared to previous region-controlled text rendering methods, our approach demonstrates significant improvements in visual quality and text

Language	EasyText (pre)		EasyText (fine)		Glyph-ByT5-v2	
	Chars	Sent	Chars	Sent	Chars	Sent
French	**0.9867**	0.8552	0.9673	0.7500	0.9812	**0.8625**
German	0.9818	**0.8333**	0.9732	0.7619	**0.9828**	0.7976
Korean	0.9341	0.7125	0.9203	0.6713	**0.9441**	**0.7437**
Japanese	**0.9265**	**0.7179**	0.9194	0.6562	0.9059	0.6187
Italian	**0.9740**	**0.9125**	0.9638	0.8571	0.9385	0.8333
Thai	**0.9628**	**0.7443**	0.9334	0.6500	✗	✗
Vietnamese	**0.9605**	**0.7312**	0.9401	0.6474	✗	✗
Greek	**0.9702**	**0.7685**	0.9360	0.6875	✗	✗

Table 4: Multilingual text generation precision with our unified and generalized model, compared to Glyph-SDXL-v2.

Metric	Ours (pre)	Ours (ft)	Glyph-v2	SD3.5	FLUX
CLIPScore	0.3318	0.3486	0.3140	**0.3519**	0.3348
OCR Acc. (%)	84.32	**88.72**	82.33	79.33	76.09
Evaluation by GPT-4o					
Image Aesthetics	81.58	83.86	75.86	**84.58**	83.61
Text Aesthetics	65.14	**73.79**	65.28	72.06	71.90
Text Quality	84.58	**90.66**	86.25	85.79	86.95
Text-Image Fusing	74.48	**81.28**	74.80	81.12	78.66

Table 5: Quantitative evaluation of text-to-image rendering across diverse metrics and GPT-4o-based assessments. **Ours (pre)** and **Ours (ft)** denote the pretrained and finetuned versions of our model, respectively.

fidelity, with better visual-text integration, higher OCR accuracy, and enhanced aesthetic coherence (shown in Fig. 3). Additionally, unlike commercial text-to-image models, our method allows for more precise spatial control over rendered text, particularly in generating multiple paragraphs of relatively long text (e.g., 4–5 paragraphs with around 20 characters each) while maintaining consistent layouts. We further evaluate the model's ability to handle challenging text generation scenarios that are typically difficult for existing methods. Specifically, our approach generalizes well to unseen characters, slanted or non-rectilinear text regions, and long text sequences, while maintaining structural consistency and legibility. (refer to Appendix Section C for details)

4.4 Ablation Studies

To validate the effectiveness of using diverse fonts in synthetic pretraining, we conducted a controlled experiment where both the target and condition images were rendered using the same standard font. Despite high initial precision, this setup captures only direct shape mappings and fails to generalize after real-world fine-tuning. (refer to Appendix Section E) This highlights the importance of font diversity—rendering targets with multiple fonts while keeping the condition font fixed—for learning robust, transferable representations. We compare our approach with three ablated variants: (1) removing position alignment, (2) removing the condition input, and (3) using the original FLUX model. As shown in Fig. 5, incorporating position mapping substantially improves spatial precision and text accuracy, particularly in multi-text scenarios. In comparison, when removing the condition input, the target rendered text is instead provided in the prompt(in our full method, rendered text content is given via the condition input). This demonstrates that our fine-tuned model effectively retains the strong generation capability of the base FLUX model.

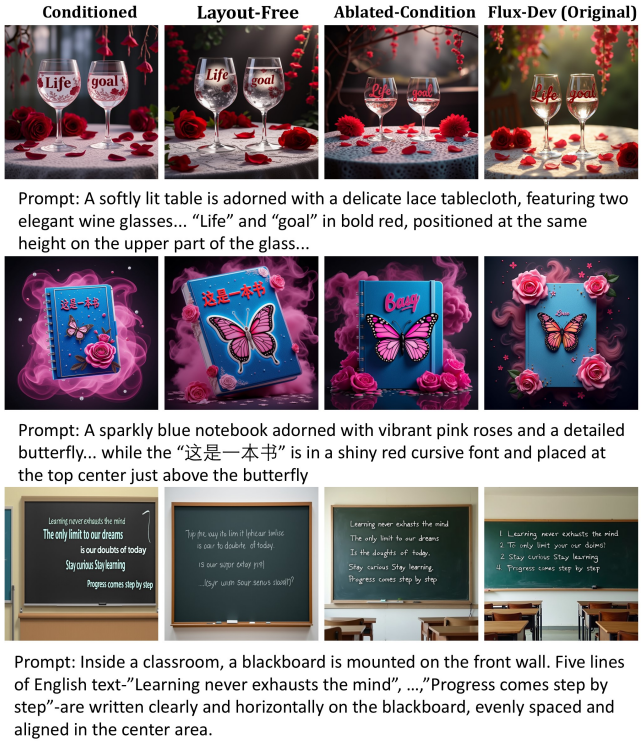

Figure 5: Ablation study comparing the full conditioned method with: (1) layout-free (2) without condition inputs, and (3) original FLUX-Dev. <sks1>, <sks2> triggers are used to replace rendered text in the first two cases.

5 Limitation

The Implicit Character Position Alignment mechanism is less effective when character positions are substantially overlapping, occasionally leading to reduced rendering accuracy. In addition, training across multiple scripts results in confusion between simple but visually similar characters from different writing systems. These cases are infrequent but consistently observed.

6 Conclusion

In this work, we propose EasyText, a diffusion-based framework for multilingual and controllable text generation in text-to-image synthesis. EasyText learns to mimic glyph features and supports precise and flexible text placement through implicit character alignment. With a two-stage training strategy, the method significantly reduces reliance on real multilingual data while maintaining high rendering accuracy. It also demonstrates strong visual-text integration, effectively embedding text into complex scenes.

References

Azadi, S.; Fisher, M.; Kim, V. G.; Wang, Z.; Shechtman, E.; and Darrell, T. 2018. Multi-content gan for few-shot font style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7564–7573.

Bar-Tal, O.; Chefer, H.; Tov, O.; Herrmann, C.; Paiss, R.; Zada, S.; Ephrat, A.; Hur, J.; Liu, G.; Raj, A.; et al. 2024. Lumiere: A space-time diffusion model for video generation. In *SIGGRAPH Asia 2024 Conference Papers*, 1–11.

Brooks, T.; Holynski, A.; and Efros, A. A. 2023. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 18392–18402.

ByteDance. 2024. Jimeng AI 2.1. https://jimeng.jianying.com/. Accessed July 10, 2025.

Cha, J.; Chun, S.; Lee, G.; Lee, B.; Kim, S.; and Lee, H. 2020. Few-shot compositional font generation with dual memory. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*, 735–751. Springer.

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

Chen, J.; Huang, Y.; Lv, T.; Cui, L.; Chen, Q.; and Wei, F. 2024. Textdiffuser-2: Unleashing the power of language models for text rendering. In *European Conference on Computer Vision*, 386–402. Springer.

Chen, J.; Yu, J.; Ge, C.; Yao, L.; Xie, E.; Wu, Y.; Wang, Z.; Kwok, J.; Luo, P.; Lu, H.; et al. 2023b. Pixart: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*.

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

Esser, P.; Kulal, S.; Blattmann, A.; and ... 2024. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. In *Forty-first International Conference on Machine Learning (ICML)*.

Feng, Z.; Guo, Q.; Xiao, X.; Xu, R.; Yang, M.; and Zhang, S. 2025. Unified Video Generation via Next-Set Prediction in Continuous Domain. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19427–19438.

Feng, Z.; and Zhang, S. 2023. Efficient vision transformer via token merger. *IEEE Transactions on Image Processing*, 32: 4156–4169.

Feng, Z.; and Zhang, S. 2024. Evolved Hierarchical Masking for Self-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Gao, Y.; Gong, L.; Guo, Q.; Hou, X.; Lai, Z.; Li, F.; Li, L.; Lian, X.; Liao, C.; Liu, L.; et al. 2025. Seedream 3.0 Technical Report. *arXiv preprint arXiv:2504.11346*.

Guo, H.; Zeng, B.; Song, Y.; Zhang, W.; Zhang, C.; and Liu, J. 2025. Any2AnyTryon: Leveraging Adaptive Position Embeddings for Versatile Virtual Clothing Tasks. *arXiv preprint arXiv:2501.15891*.

He, H.; Chen, X.; Wang, C.; Liu, J.; Du, B.; Tao, D.; and Yu, Q. 2024. Diff-font: Diffusion model for robust one-shot font generation. *International Journal of Computer Vision*, 132(11): 5372–5386.

Hertz, A.; Mokady, R.; Tenenbaum, J.; Aberman, K.; Pritch, Y.; and Cohen-Or, D. 2022. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.

Huang, S.; Song, Y.; Zhang, Y.; Guo, H.; Wang, X.; Shou, M. Z.; and Liu, J. 2025. PhotoDoodle: Learning Artistic Image Editing from Few-Shot Pairwise Data. *arXiv preprint arXiv:2502.14397*.

Ideogram Inc. 2025. Ideogram v3.0. https://ideogram.ai/. Accessed July 10, 2025.

Jiang, B.; Yuan, Y.; Bai, X.; Hao, Z.; Yin, A.; Hu, Y.; Liao, W.; Ungar, L.; and Taylor, C. J. 2025. ControlText: Unlocking Controllable Fonts in Multilingual Text Rendering without Font Annotations. *arXiv preprint arXiv:2502.10999*.

Kolors Team. 2024. Kolors 2.0. https://github.com/Kwai-Kolors/Kolors. Accessed July 10, 2025.

Labs, B. F. 2024. Flux. https://github.com/black-forest-labs/flux. Accessed July 10, 2025.

Li, C.; Jiang, C.; Liu, X.; Zhao, J.; and Wang, G. 2024a. JoyType: A Robust Design for Multilingual Visual Text Creation. *arXiv preprint arXiv:2409.17524*.

Li, M.; Yang, T.; Kuang, H.; Wu, J.; Wang, Z.; Xiao, X.; and Chen, C. 2024b. ControlNet++: Improving Conditional Controls with Efficient Consistency Feedback: Project Page: liming-ai. github. io/-ControlNet_Plus_Plus. In *European Conference on Computer Vision*, 129–147. Springer.

Liang, Z.; Xu, Y.; Hong, Y.; Shang, P.; Wang, Q.; Fu, Q.; and Liu, K. 2024. A Survey of Multimodel Large Language Models. In *Proceedings of the 3rd International Conference on Computer, Artificial Intelligence and Control Engineering*, 405–409.

Liu, Z.; Liang, W.; Liang, Z.; Luo, C.; Li, J.; Huang, G.; and Yuan, Y. 2024a. Glyph-byt5: A customized text encoder for accurate visual text rendering. In *European Conference on Computer Vision*, 361–377. Springer.

Liu, Z.; Liang, W.; Zhao, Y.; Chen, B.; Liang, L.; Wang, L.; Li, J.; and Yuan, Y. 2024b. Glyph-byt5-v2: A strong aesthetic baseline for accurate multilingual visual text rendering. *arXiv preprint arXiv:2406.10208*.

Ma, J.; Deng, Y.; Chen, C.; Du, N.; Lu, H.; and Yang, Z. 2025a. Glyphdraw2: Automatic generation of complex glyph posters with diffusion models and large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 5955–5963.

Ma, J.; Zhao, M.; Chen, C.; Wang, R.; Niu, D.; Lu, H.; and Lin, X. 2023. Glyphdraw: Seamlessly rendering text with intricate spatial structures in text-to-image generation. *arXiv preprint arXiv:2303.17870*.

Ma, Y.; Feng, K.; Hu, Z.; Wang, X.; Wang, Y.; Zheng, M.; He, X.; Zhu, C.; Liu, H.; He, Y.; et al. 2025b. Controllable Video Generation: A Survey. *arXiv preprint arXiv:2507.16869*.

Ma, Y.; He, Y.; Wang, H.; Wang, A.; Shen, L.; Qi, C.; Ying, J.; Cai, C.; Li, Z.; Shum, H.-Y.; et al. 2025c. Follow-Your-Click: Open-domain Regional Image Animation via Motion Prompts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 6018–6026.

Ma, Y.; Liu, H.; Wang, H.; Pan, H.; He, Y.; Yuan, J.; Zeng, A.; Cai, C.; Shum, H.-Y.; Liu, W.; et al. 2024. Follow-your-emoji: Fine-controllable and expressive freestyle portrait animation. In *SIGGRAPH Asia 2024 Conference Papers*, 1–12.

Ma, Y.; Liu, Y.; Zhu, Q.; Yang, A.; Feng, K.; Zhang, X.; Li, Z.; Han, S.; Qi, C.; and Chen, Q. 2025d. Follow-Your-Motion: Video Motion Transfer via Efficient Spatial-Temporal Decoupled Finetuning. *arXiv preprint arXiv:2506.05207*.

Ma, Y.; Yan, Z.; Liu, H.; Wang, H.; Pan, H.; He, Y.; Yuan, J.; Zeng, A.; Cai, C.; Shum, H.-Y.; et al. 2025e. Follow-your-emoji-faster: Towards efficient, fine-controllable, and expressive freestyle portrait animation. *arXiv preprint arXiv:2509.16630*.

Mou, C.; Wang, X.; Xie, L.; Wu, Y.; Zhang, J.; Qi, Z.; and Shan, Y. 2024. 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*, volume 38, 4296–4304.

OpenAI. 2024. GPT-4o. https://chatgpt.com/. Accessed July 10, 2025.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4195–4205.

Podell, D.; English, Z.; Lacey, K.; Blattmann, A.; Dockhorn, T.; Müller, J.; Penna, J.; and Rombach, R. 2023. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*.

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

Shi, W.; Song, Y.; Rao, Z.; Zhang, D.; Liu, J.; and Zou, X. 2025. WordCon: Word-level Typography Control in Scene Text Rendering. *arXiv preprint arXiv:2506.21276*.

Shi, W.; Song, Y.; Zhang, D.; Liu, J.; and Zou, X. 2024. FonTS: Text Rendering with Typography and Style Controls. *arXiv preprint arXiv:2412.00136*.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y.; Chen, D.; and Shou, M. Z. 2025. LayerTracer: Cognitive-Aligned Layered SVG Synthesis via Diffusion Transformer. *arXiv preprint arXiv:2502.01105*.

Song, Y.; Huang, S.; Yao, C.; Ye, X.; Ci, H.; Liu, J.; Zhang, Y.; and Shou, M. Z. 2024. ProcessPainter: Learn Painting Process from Sequence Data. *arXiv preprint arXiv:2406.06062*.

Song, Y.; Liu, C.; and Shou, M. Z. 2025. MakeAnything: Harnessing Diffusion Transformers for Multi-Domain Procedural Sequence Generation. *arXiv preprint arXiv:2502.01572*.

Song, Y.; Shao, X.; Chen, K.; Zhang, W.; Jing, Z.; and Li, M. 2023. Clipvg: Text-guided image manipulation using differentiable vector graphics. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 2312–2320.

Song, Y.; and Zhang, Y. 2022. CLIPFont: Text Guided Vector WordArt Generation. In *BMVC*, 543.

Tan, Z.; Liu, S.; Yang, X.; Xue, Q.; and Wang, X. 2024. Ominicontrol: Minimal and universal control for diffusion transformer. *arXiv preprint arXiv:2411.15098*.

Tang, L.; Cai, Y.; Liu, J.; Hong, Z.; Gong, M.; Fan, M.; Han, J.; Liu, J.; Ding, E.; and Wang, J. 2022. Few-shot font generation by learning fine-grained local styles. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7895–7904.

Thamizharasan, V.; Liu, D.; Agarwal, S.; Fisher, M.; Gharbi, M.; Wang, O.; Jacobson, A.; and Kalogerakis, E. 2024. Vecfusion: Vector font generation with diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7943–7952.

Tuo, Y.; Geng, Y.; and Bo, L. 2024. AnyText2: Visual Text Generation and Editing With Customizable Attributes. *arXiv preprint arXiv:2411.15245*.

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

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wan, C.; Luo, X.; Cai, Z.; Song, Y.; Zhao, Y.; Bai, Y.; He, Y.; and Gong, Y. 2024. Grid: Visual layout generation. *arXiv preprint arXiv:2412.10718*.

Wang, H.; Xu, Y.; Li, Y.; Li, J.; Zhang, C.; Wang, J.; Yang, K.; and Chen, Z. 2025. RepText: Rendering Visual Text via Replicating. *arXiv preprint arXiv:2504.19724*.

Wang, Q.; Bai, X.; Wang, H.; Qin, Z.; Chen, A.; Li, H.; Tang, X.; and Hu, Y. 2024a. Instantid: Zero-shot identity-preserving generation in seconds. *arXiv preprint arXiv:2401.07519*.

Wang, R.; Guo, H.; Liu, J.; Li, H.; Zhao, H.; Tang, X.; Hu, Y.; Tang, H.; and Li, P. 2024b. Stablegarment: Garment-centric generation via stable diffusion. *arXiv preprint arXiv:2403.10783*.

Wu, L.; Zhang, C.; Liu, J.; Han, J.; Liu, J.; Ding, E.; and Bai, X. 2019. Editing text in the wild. In *Proceedings of the 27th ACM international conference on multimedia*, 1500–1508.

Yang, Y.; Gui, D.; Yuan, Y.; Liang, W.; Ding, H.; Hu, H.; and Chen, K. 2023. Glyphcontrol: glyph conditional control for visual text generation. *Advances in Neural Information Processing Systems*, 36: 44050–44066.

Zhang, L.; Chen, X.; Wang, Y.; Lu, Y.; and Qiao, Y. 2024a. Brush your text: Synthesize any scene text on images via diffusion model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 7215–7223.

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

Zhang, Y.; Song, Y.; Liu, J.; Wang, R.; Yu, J.; Tang, H.; Li, H.; Tang, X.; Hu, Y.; Pan, H.; et al. 2024b. Ssr-encoder: Encoding selective subject representation for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8069–8078.

Zhang, Y.; Song, Y.; Yu, J.; Pan, H.; and Jing, Z. 2024c. Fast personalized text to image synthesis with attention injection. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6195–6199. IEEE.

Zhang, Y.; Wei, L.; Zhang, Q.; Song, Y.; Liu, J.; Li, H.; Tang, X.; Hu, Y.; and Zhao, H. 2024d. Stable-Makeup: When Real-World Makeup Transfer Meets Diffusion Model. *arXiv preprint arXiv:2403.07764*.

Zhang, Y.; Yuan, Y.; Song, Y.; Wang, H.; and Liu, J. 2025. Easycontrol: Adding efficient and flexible control for diffusion transformer. *arXiv preprint arXiv:2503.07027*.

Zhang, Y.; Zhang, Q.; Song, Y.; and Liu, J. 2024e. Stable-Hair: Real-World Hair Transfer via Diffusion Model. *arXiv preprint arXiv:2407.14078*.

Zhao, Y.; and Lian, Z. 2023. Udifftext: A unified framework for high-quality text synthesis in arbitrary images via character-aware diffusion models. *arXiv preprint arXiv:2312.04884*.