ControlText: Unlocking Controllable Fonts in Multilingual Text Rendering without Font Annotations

Anonymous ACL submission

Abstract

This work demonstrates that diffusion models can achieve *font-controllable* multilingual text rendering using just raw images without font label annotations. Visual text rendering remains a significant challenge. While recent methods condition diffusion on glyphs, it is impossible to retrieve exact font annotations from large-scale, real-world datasets, which prevents user-specified font control. To address this, we propose a data-driven solution that integrates the conditional diffusion model with a text segmentation model, utilizing segmentation masks to capture and represent fonts *in pixel space* in a *self-supervised* manner, thereby eliminating the need for any ground-truth labels and enabling users to customize text rendering with any multilingual font of their choice. The experiment provides a proof of concept of our algorithm in zero-shot text and font editing across diverse fonts and languages, providing valuable insights for the community and industry toward achieving generalized visual text rendering.

1 Introduction

Diffusion models have become the dominant paradigm in image generation (Ho et al., 2020; Saharia et al., 2022; Zhang et al., 2023b; Rombach et al., 2022; Betker et al., 2023; Ramesh et al., 2022; Esser et al., 2024), because of their iterative denoising process that allows fine-grained image synthesis. While these models effectively capture data distributions of photorealistic or artistic images, they still fall short in generating high-fidelity text. Rendering text in images is inherently more challenging as it requires precise knowledge of the geometric alignment among strokes, the arrangement of letters as words, the legibility across varying fonts, sizes, and styles, and the integration of text into visual backgrounds. At the same time, humans are more sensitive to minor errors in text, such as a missing character or an incorrectly shaped letter, compared to natural elements in a visual scene that allow for a much higher degree of variation.

Increasing attention has been paid to visual text rendering (Bai et al., 2024; Han et al., 2024; Li and Lian, 2024) due to its high user demands. Instead of relying solely on diffusion models to remember exactly how to render text, recent research is starting to embed the visual attributes of texts, such as glyphs (Tuo et al., 2023; Liu et al., 2024; Ma et al., 2024; Yang et al., 2024b), as input conditions to diffusion models. However, it is still difficult for users to specify the desired font in the open world, and there remain open challenges that burden the development of font-controllable text rendering:

- No ground-truth font label annotation is available in the massive training dataset, while synthetic images often fail to accurately mimic subtle details that appear in reality.
- There are numerous fonts available in the open world, but many fonts with different names are very similar which confounds evaluation.
- Users like visual designers may want to explore different fonts during their design process, even creating novel fonts of their own.
- State-of-the-art image generation models are either closed-sourced (Betker et al., 2023; Midjourney) or users still cannot edit texts or fonts without altering other parts of the image.

This work aims to address the above challenges. To summarize our contributions, we introduce the simplest and, to our best knowledge, one of the few (Ma et al., 2024; Liu et al., 2024) open-source methods for rendering visual text with user-controllable fonts. We provide code in the hope that others can draw inspiration from the underlying **data-driven algorithm** and benefit from the **simplicity in the self-supervised training**.

Figure 1: Examples of real-world test images with text generated by ControlText in various fonts and languages. Each row presents both the rendered images and the textual part of the corresponding glyph controls that provide the text and the intricate font information in pixel space.

We also provide the community a comprehensive dataset for font-aware glyph controls collected from diverse real-world images. We further propose a **quantitative evaluation metrics for handling fuzzy fonts in the open world.** Experimental results demonstrate that our method, ControlText, as a text and font editing model, facilitates a human-in-the-loop process to generate multilingual text with user-controllable fonts in a zero-shot manner.

2 Related Work

Generation from Prompts or Text Embeddings Text-to-image generation (Zhang et al., 2023a; Bie et al., 2023) has advanced significantly in recent years, leveraging conditional latent diffusion models (Ho et al., 2020; Rombach et al., 2022; Zhang et al., 2023b). Foundational image generation models (Ramesh et al., 2021; Betker et al., 2023; Midjourney; Saharia et al., 2022; AI; Esser et al., 2024; Yang et al., 2024a; Zhao et al., 2023; Hoe et al., 2024; Sun et al., 2025; Chang et al., 2022) have achieved remarkable progress in creating high-quality photo-realistic and artistic images.

Despite these advancements, visual text rendering (Bai et al., 2024; Han et al., 2024; Li and Lian, 2024) continues to pose significant challenges. Several algorithms rely on text embeddings from user prompts or captions to control the diffusion process, such as TextDiffuser (Chen et al., 2024b), TextDiffuser2 (Chen et al., 2025), and DeepFloyd's IF (DeepFloyd-Lab, 2023). Li et al. (2024b) utilizes intermediate features from OCR (Du et al., 2020) as text embeddings, Liu et al. (2022); Wang et al. (2024b); Choi et al. (2024) take one step deeper into the character level, and TextHarmony (Zhao et al., 2024b) queries a fine-tuned vision-language model to generate embeddings from images and captions.

Generation from Glyphs The majority of algorithms rely on visual glyphs - pixel-level representations that describe the appearance of texts - to guide the generation process. However, massive real-world images in the training dataset often lack ground-truth font annotations, while it is challenging to extract exact font labels from any real-world photos. **Consequently, these approaches utilize a fixed standard font to render the texts on their glyph controls.** For instance, GlyphControl (Yang et al., 2024b) and GlyphDraw (Ma et al., 2023) render text detected by OCR models, with the former introducing an additional Glyph ControlNet to regulate the decoder of the original ControlNet model (Zhang et al., 2023b). TextMaster (Wang et al., 2024a), DiffUTE (Chen et al., 2024a), and AnyTrans (Qian et al., 2024) aim to maintain uniform fonts within the same image. Choi et al. (2024) explores design translation for artistic char-

acters. Similarly, Zhang et al. (2024) employs randomly chosen fonts and a Canny edge detector. Several approaches also incorporate automatic textual layout generation (Tuo et al., 2023; Zhu et al., 2024; Chen et al., 2024a; Li et al., 2024c; Seol et al., 2025; Lakhanpal et al., 2024; Paliwal et al., 2024b; Zhao et al., 2024a). For example, (Zhu et al., 2024) uses a vision-language model to identify suitable layouts, DiffUTE (Chen et al., 2024a) leverages language models and OCR detectors to propose bounding boxes, and (Seol et al., 2025) employs HTML codes. In constrast, we concentrate on human-in-the-loop local text editing rather than automatic global layout generation. Our work also eliminates the need for any large language or multimodal models, streamlining the pipeline.

Our work builds upon the codebase of AnyText (Tuo et al., 2023), a glyph-based algorithm that trains a base ControlNet (Zhang et al., 2023b) model to render visual texts. Since ground-truth font annotations are not accessible from real-world images, glyphs are generated in a fixed standard font, leaving the model to infer an appropriate font.

Font-Controllable Generation Fewer recent works are more closely related to ours in enabling controllable fonts (Tuo et al., 2024; Ma et al., 2024; Li et al., 2024a; Shi et al., 2024; Paliwal et al., 2024a; Liu et al., 2025). **However, none of these works provide a quantitative evaluation metric to assess the generated fonts in open-world settings.** AnyText2 (Tuo et al., 2024) is a concurrent work developed by the authors of AnyText (Tuo et al., 2023). We share a similar architecture, but unlike (Tuo et al., 2024), we eliminate its use of lengthy language prompts in the inputs, separate models to support different languages, and the trainable OCR model to encode the font features. Instead, we use OCR solely to filter out low-quality glyph controls. Tuo et al. (2024) is also not yet open-sourced at the time of our submission.

Glyph-ByT5 (Liu et al., 2025) and its v2 (Liu et al., 2024) also support font control. However, unlike (Liu et al., 2025), which *pre-specifies font labels for each language* and focuses on improving language understanding, we assume that text rendering is independent of the semantic meanings; in our approach, only pixel-level controls matter. GlyphDraw2 (Ma et al., 2024) learns font features through cross-attention between glyph features and hidden variables within the image while fine-tuning a language model to generate font layouts. Unfortunately, it lacks quantitative assessment on fonts, relying on human evaluations. JoyType (Li et al., 2024a) focuses on e-commerce product images in non-Latin languages using 10 *pre-specified fonts with synthetic images* and additional vision-language models. It employs an OCR model for font perceptual losses, while we assume OCR to be font-agnostic. FonTS (Shi et al., 2024) uses additional reference images as font style controls and *requires users to select fonts from pre-specified integer font labels*. Similarly, CustomText (Paliwal et al., 2024a) relies on specific font names provided in extended user prompts and an additional training datasets tailored for smaller fonts. While Liu et al. (2024) emphasizes visual aesthetics and highlights challenges with small fonts, our work centers on font editing and share the same perspective that smaller fonts pose greater difficulties. However, we show that zooming into localized regions for text editing can already improve the quality of smaller text. **Different from those works, we eliminate the need for pre-specified fonts with unique font names or special font tokens** (Liu et al., 2024; Shi et al., 2024; Paliwal et al., 2024a) in user prompts, making our method **generalizable to unseen languages and fonts**.

3 Technical Approach

We envision this method being used as a modular plug-in for existing text-to-image generation frameworks. It works with images generated by any base models or actual photos. For instance, when incorrect text is generated, or the user wants to replace some text or modify its font, our algorithm can be specifically targeted to these localized regions without altering remaining parts in images. By leveraging a human-in-the-loop approach, the model aims to render controllable visual text within the user-specified region, perform background inpainting, and blend the modified region back into the original image, regardless of its original size.

3.1 Data-Driven Insights

ControlText employs a data-driven approach to unlock visual text rendering with user-controllable fonts without relying on complex architectural designs, aiming to avoid pitfalls associated with the bitter lesson (Sutton, 2019) in foundation models. We feed the model a massive and diverse amount of unsupervised glyphs, each containing detailed font features in pixel space, and train the diffusion

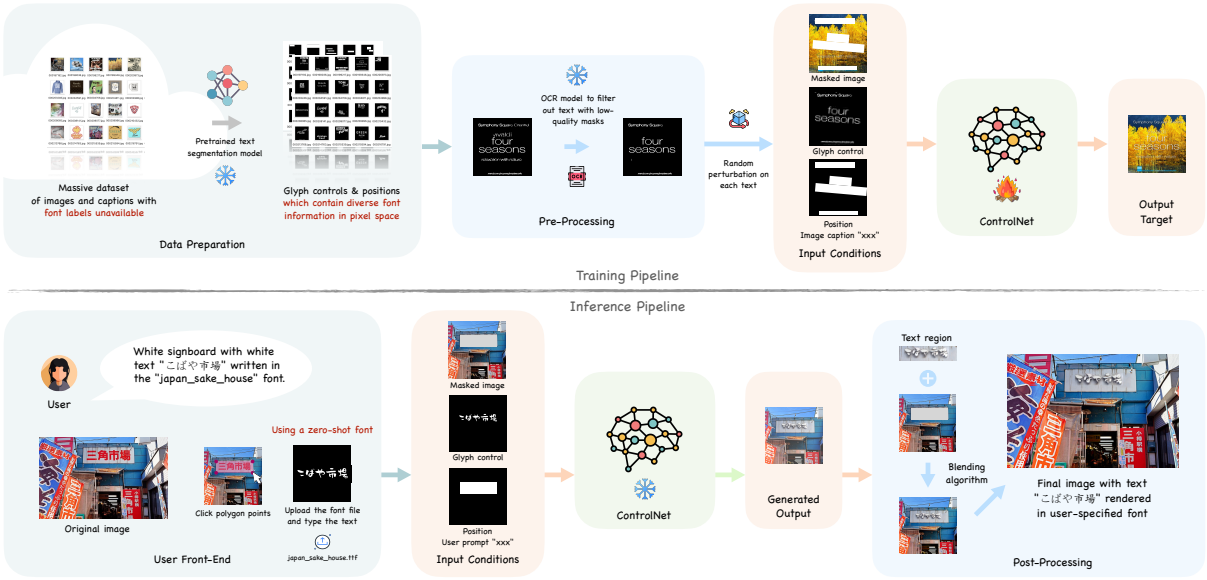

Figure 2: System overview. It consists of two parts (1) Training pipeline: text segmentation masks are extracted as glyph controls from a large image dataset without ground-truth font annotations. Low-quality masks are filtered out using an OCR model, and random perturbations are applied to prevent the model from overfitting to exact pixel locations of the glyphs. (2) Inference pipeline: users upload images, specify text regions, and provide any desired font file through the user front-end. The model generates an image patch with the rendered text, which is then seamlessly blended into the original image. Throughout this figure, models marked with a fire icon indicate trainable weights, while those marked with a snowflake icon are frozen.

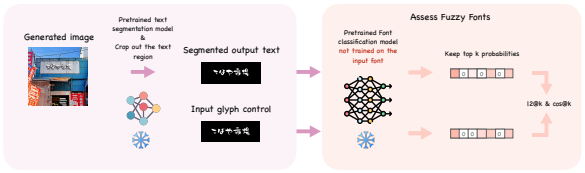

Figure 3: Evaluation pipeline: the cropped regions of the generated text and the input glyph are processed by a pretrained font classification model, which may not have seen the user-specified font. The proposed $l_2@k$ and $\cos@k$ metrics for fuzzy fonts assume that similar fonts have similar output probability vectors, while we retain only top-k values while zeroing out the rest.

process to reconstruct target images following the explicit pixel-level hints in these input glyphs.

The key insight is that during training, the model learns to leverage pixel-level controls provided by the glyphs as a shortcut for generating text within images. Importantly, the glyphs no longer need to use a standard font, but can mimic any font appear in the target images. Unlike traditional methods, no annotations of font labels are required; the only guidance comes from how the text appears in pixel space as depicted by the input glyphs.

In inference, **the model should have seen a diverse set of glyphs during training, including intricate font features represented by pixel details near the textual edges in the glyphs.** With this information, it can render unseen languages or unfamiliar text without requiring prior knowledge of how to write the text from scratch, how to arrange individual letters or characters, or understanding their semantic meaning. **The model just treats text as a collection of pixels rather than linguistic entities.** This self-supervised data-driven approach not only enhances the model's generalizability to open-world scenarios, but also ensures scalability when more image data, computation, and larger base models become available.

3.2 Training Pipeline

3.2.1 Collection of Font-Aware Glyph Controls

Our training pipeline begins with the collection of glyph controls by performing text segmentation on images. We use TexRNet (Xu et al., 2021) as our text segmentation algorithm to identify text regions and provide fine-grained masks, preserving intricate features of different fonts in pixel space, It also provides bounding boxes that will serve as position controls (Tuo et al., 2023). The segmentation algorithm is a pre-trained deep-learning-based model, so it may occasionally miss masks for certain letters or parts of letters. As a result, we introduce an

OCR model, specifically PaddleOCR-v3 (Du et al., 2020), into the pipeline following the segmentation process. The OCR model validates the detected text by filtering out masks that fail to meet our quality criteria, regardless of the font: (1) an OCR confidence score no lower than 0.8. (2) an edit distance no greater than 20% of the string length. This step ensures that only high-quality segmentation masks are retained as glyph controls.

3.2.2 Perspective Distortion

Segmentation masks, even after quality filtering, are not directly usable as glyph controls. In real-world scenarios, **users are unlikely to specify the exact locations of text or precisely align the text with the background**. To address this issue, we apply random perspective transformations to the collected glyph images, introducing slight translations and distortions to the text, without affecting the fonts. Specifically, we add random perturbations to the four corner points of the text's bounding box, with the perturbation upper-bounded by ϵ pixels.

We then compute a homography matrix $\boldsymbol{M} \in \mathbb{R}^{3\times3}$ that maps the original text region to a slightly distorted view. **This design ensures that the diffusion model does not rigidly replicate the exact pixel locations of the glyphs** but instead learns to adaptively position the text in a way that best integrates with the output image.

3.2.3 Main Training Process

The diffusion process builds upon AnyText (Tuo et al., 2023), leveraging ControlNet (Zhang et al., 2023b) as the base model. As shown in Figure 2, the model takes the following five inputs during training. We expect the training dataset to consist of images containing text, captions, and polygons for the text region. The text and polygons can be automatically extracted using an OCR algorithm.

- Font-aware glyph control $\boldsymbol{c}_g \in \mathbb{R}^{n\times n}$: A binary mask representing the text and its font features in pixel space.
- Position control $\boldsymbol{c}_p \in \mathbb{R}^{n\times n}$: A binary mask for the bounding box of the text region. We restrict ourselves to square local image regions.
- Masked image $\boldsymbol{c}_m \in \mathbb{R}^{n\times n\times3}$: An RGB image normalized to the range $[-1, 1]$, where the region within the box position $\boldsymbol{c}_p$ is masked to 0. Every other pixel is identical to the target image $\boldsymbol{I} \in \mathbb{R}^{n\times n\times3}$.
- Image caption $\boldsymbol{c}_l$: We adopt the same handling approach in Tuo et al. (2023), except for using our own $\boldsymbol{c}_g$. We empirically observe that image captions are not crucial for this work.
- Random noise input $\boldsymbol{x}_T \in \mathbb{R}^{m\times m\times d}$ in the embedding space to initialize the reverse denoising process (Ho et al., 2020).

The model outputs a denoised tensor $\boldsymbol{x}_T \in \mathbb{R}^{m\times m\times d}$ after T timesteps, which can be reconstructed back to an image $\hat{\boldsymbol{I}} \in \mathbb{R}^{n\times n\times3}$. In the expressions above, n denotes the edge length of the image, m represents the spatial resolution of the hidden features in the latent diffusion (Ho et al., 2020), and d represents the number of channels.

We concatenate all input conditions as $\boldsymbol{c}$ and perform the following reverse denoising process:

$$\boldsymbol{c} = \varphi(\mathrm{cat}[\xi_g(\boldsymbol{c}_g), \xi_p(\boldsymbol{c}_p), \xi_m(\boldsymbol{c}_m)]) \quad (1)$$

$$p_\theta\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{c}\right) = \mathcal{N}\left(\boldsymbol{x}_{t-1}; \mu_\theta\left(\boldsymbol{x}_t, t, \boldsymbol{c}\right), \Sigma_\theta(t)\right) \quad (2)$$

where each ξ is some convolutional layers that transform the input to $\mathbb{R}^{m\times m\times d}$, φ is another fusion layer, p_θ is the probabilistic model that predicts the distribution of a less noisy image $\boldsymbol{x}_{t-1}$ from $\boldsymbol{x}_t$ with $t \in [0, T]$, and μ_θ and Σ_θ are the mean and variance of the normal distribution $\mathcal{N}$. We follow the same training losses in AnyText (Tuo et al., 2023) to train this diffusion model.

3.3 Inference Pipeline

Our philosophy is to design a more streamlined training pipeline that is easily scalable to larger open-world datasets, while shifting additional steps to inference time to provide users with greater control and flexibility as needed.

3.3.1 Main Generation Process

The reversed denoising process takes the same set of inputs outlined in Section 3.2.3. However, unlike training where the glyph control $\boldsymbol{c}_g$ is extracted using the text segmentation model $\mathcal{M}$, it is now provided directly by the user.

On the user front-end, the required inputs include the original image $\boldsymbol{I}$ with a short caption $\boldsymbol{c}_l$, the desired text t, the font f (which can be uploaded as a font file), and the polygon points $\boldsymbol{p}$ selected on $\boldsymbol{I}$ to define the region where the text will be rendered. To streamline the process, the pipeline automatically converts polygon points $\boldsymbol{p}$ into the position control $\boldsymbol{c}_p$, generates the masked image

$\boldsymbol{c}_m$, and converts text t into the font-aware glyph control $\boldsymbol{c}_g$. Users are allowed to type multiple lines of text, possibly in different languages, fonts, or orientations, in a single $\boldsymbol{c}_p$, $\boldsymbol{c}_g$, and $\boldsymbol{c}_m$.

Finally, the reverse denoising process is run over t timesteps following the same Equations 1-2 to generate the output image $\boldsymbol{x}_0$, whose region within the polygon mask $\boldsymbol{c}_p$ is will be blended into the original image $\boldsymbol{I}$ using normal seamless cloning or other blending algorithms. This completes the generation process, and the next two subsections illustrate optional steps that can be applied as needed.

3.3.2 Inpainting Before Editing

When editing text in an image, the mask $\boldsymbol{c}_m$ must encompass all the old text in the background. However, this mask could be larger than the size of the new text t in the new font f, particularly when a narrower font is selected. Larger masks may introduce additional text rendered in the output image not specified in the glyph control $\boldsymbol{c}_g$. To address this challenge, we minimize the mask size to be just large enough to fit the text t in the new font f. Following recommendations in (Li et al., 2024c), we utilize an off-the-shelf inpainting model (Razzhigaev et al., 2023) to erase the original text. After inpainting, a new polygon $\hat{\boldsymbol{p}}$ is automatically tightened from $\boldsymbol{p}$ to match the new text.

3.3.3 Small Textual Regions

Handling smaller text remains a challenge (Liu et al., 2024; Paliwal et al., 2024a), as the diffusion process operates in the embedding space with potential information loss. To address this, we simply zoom into the text region specified by the user and interpolate it to the input size of the diffusion model. Finally, we blend the generated region with the original image $\boldsymbol{I}$. Figure 4 includes some examples of small text rendered with high quality, demonstrating effective performance without the need for more complex algorithms or datasets.

3.4 Evaluation Metrics

3.4.1 Evaluating Text

We adopt the same evaluation metrics from AnyText (Tuo et al., 2023) to ensure that the generated text remains recognizable regardless of the font. Specifically, we utilize Sentence Accuracy (SenACC) and Node similarity with Edit Distance (NED) derived from OCR to assess text recognizability. We also employ Fréchet Inception Distance (FID) to evaluate the overall image quality.

3.4.2 Evaluating Fuzzy Fonts

Evaluating the accuracy of fonts in visual text remains an open question, as ground-truth font labels are typically unavailable in large-scale, real-world datasets. It is also the case that many fonts appear visually similar, making distinctions among them practically meaningless. **These challenges highlight the need for a new evaluation metric that can handle fuzzy fonts in an open-world scenario.** To address this, we introduce a novel evaluation framework leveraging a pre-trained font classification model $\mathcal{F}$. Specifically, we use the Google Font Classifier by Storia-AI (AI, 2025), an open-source model trained on $c = 3474$ fonts on both real and AI-generated images. Due to the large value of c, the classifier's embedding space is expected to provide meaningful representations, *even if the model may have never encountered the evaluated font before.* For example, **two fonts that look similar should have similar embeddings in this pretrained font classification model**, and vice versa. Therefore, we propose two metrics $\boldsymbol{l_2@k}$ and $\boldsymbol{\cos\ @k}$ to evaluate font fidelity in any generated images with text of any fonts.

- **Step 1 Embedding Extraction:** Both the input glyph $\boldsymbol{c}_g$ and the output image $\boldsymbol{x}_0$ are forwarded through the font classification model $\mathcal{F}$ to obtain their last-layer probabilities $\boldsymbol{p}_g, \boldsymbol{p}_x \in \mathbb{R}^c$, respectively, where c is the number of labels $\mathcal{F}$ is pretrained on. Optionally, text regions in $\boldsymbol{x}_0$ can be first isolated using a text segmentation model $\mathcal{M}$, eliminating the influence of color and background.

- **Step 2 Distance Calculation:** We retain only the top k largest values in $\boldsymbol{p}_g$ and $\boldsymbol{p}_x$, zeroing out the others, to ensure that the distance calculation focuses on the most likely k labels. It helps reduce disturbances from the accumulation of remaining insignificant values. The metric $l_2@k$ and $\cos@k$ then compute the l_2-distance and cos-distance between them.

4 Experimental Results

4.1 Experimental Setups

We finetune the ControlNet (Zhang et al., 2023b) model, with a size of around 1B pretrained by AnyText (Tuo et al., 2023), for 10 epochs using 4 NVIDIA V100 GPUs, each with 32 GB memory. We use a batch size of 6, a learning rate of 2×10^{-5}, and focus solely on inpainting masked images. The

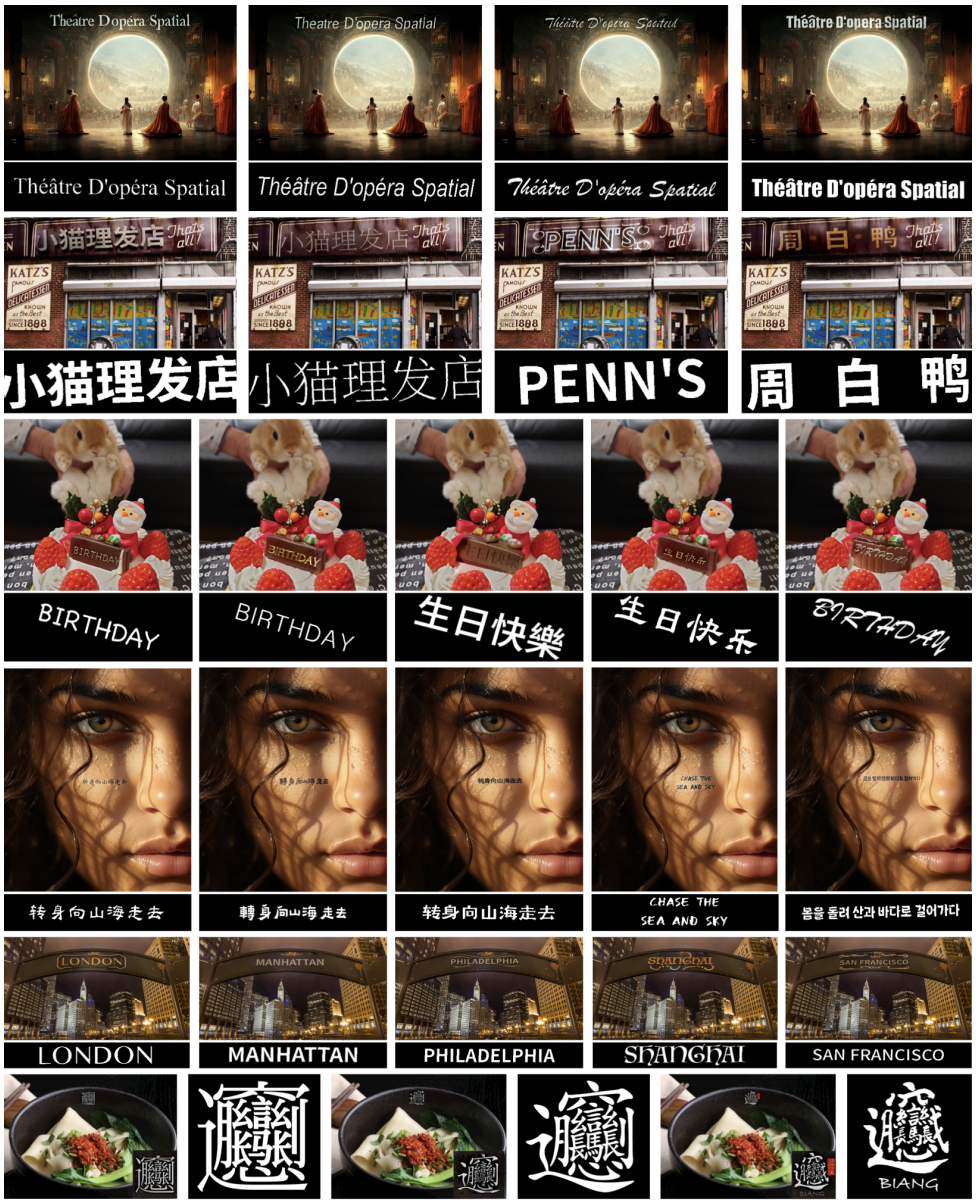

Figure 4: Continuation of Figure 1. Examples of real-world and AI-generated images with text generated by ControlText in various fonts and languages. Each row presents both the rendered images and the textual part of their glyph controls. We also try the most complex Chinese character, “biang", in the bottom row, accompanied by a zoomed-in view of the rendered character. ControlText effectively renders text with realistic integration into backgrounds while maintaining correct letters and characters in their user specified fonts.

Table 1: Results on AnyText-benchmark. “AnyText-v1.1 Font Aware" refers to AnyText (Tuo et al., 2023), but uses the same font-aware glyph controls from our work without fine-tuning as an ablation, which shows significant performance drop in text accuracy. Although with little decrease in non-Latin text accuracy, ControlText outperforms AnyText by a large margin in preserving diverse and detailed font information in input glyph controls.

English											
Methods	**Text Accuracy ↑**		**Image Quality ↓**	**Fuzzy Font Accuracy (in Distance) ↓**							
	SenACC	NED	FID	l_2@5	l_2@20	l_2@50	l_2@full	cos @5	cos @20	cos @50	cos @full
AnyText-v1.1 (Tuo et al., 2023)	0.8315	0.9081	26.18	0.3038	0.2581	0.2496	0.2466	0.1615	0.1599	0.1603	0.1614
AnyText-v1.1 Font-Aware	0.5524	0.6839	**24.95**	0.3049	0.2589	0.2497	0.2458	0.1601	0.1583	0.1587	0.1597
ControlText (ours)	**0.8345**	**0.9107**	25.93	**0.2449**	**0.1850**	**0.1750**	**0.1709**	**0.1305**	**0.1286**	**0.1291**	**0.1301**
Chinese											
Methods	**Text Accuracy ↑**		**Image Quality ↓**	**Fuzzy Font Accuracy (in Distance) ↓**							
	SenACC	NED	FID	l_2@5	l_2@20	l_2@50	l_2@full	cos @5	cos @20	cos @50	cos @full
AnyText-v1.1 (Tuo et al., 2023)	**0.8591**	**0.8284**	**27.08**	0.3103	0.2724	0.2643	0.2608	0.1521	0.1513	0.1519	0.1530
AnyText-v1.1 Font-Aware	0.5578	0.3396	27.11	0.3106	0.2799	0.2713	0.2690	0.1478	0.1472	0.1477	0.1488
ControlText (ours)	0.7867	0.7479	28.63	**0.2602**	**0.1992**	**0.1891**	**0.1848**	**0.1274**	**0.1271**	**0.1275**	**0.1286**

dataset is curated from AnyWord-3M (Tuo et al., 2023) but with our font-aware glyph. Each RGB image of size 512 by 512 has at most 5 lines of text. The dataset comprises approximately 1.39 million images in English, 1.6 million images in Chinese, and 10 thousand images in other languages. Following this, we continue training the model for another 2 epochs, turning on the textual perception loss introduced in (Tuo et al., 2023). We use AnyText-benchmark (Tuo et al., 2023) with 1000 test images in English and Chinese to show quantitative results.

4.2 Visual Results

Figures 1 and 4 showcase open-world images generated by our model. We always follow the text editing pipeline to either modify existing text or render new text. The original images $\boldsymbol{I}$ used in our experiment include both real (PIXTA, n.d.; Unsplash, n.d.; Business Insider, 2011; CNN Travel, n.d.; peterpom211, 2024; Tripadvisor, n.d.; Nipic, n.d.) and AI-generated (Wikipedia contributors, n.d.; Monks, 2023) examples. ControlText demonstrates high-fidelity text rendering, accurately preserving both the text and the font styles. It automatically render text in either flat formats or with depth and color effects based on the background, such as outward-engraved text on a shabby storefront sign on the street, a metallic board on wall, a chocolate bar, or with neon light effects at night.

We present images in multiple languages: English, French (zero-shot), traditional and simplified Chinese (including *the* most complex character *"biang"*), Japanese (including Kaomoji), and Korean, rendered in either single or multi-line formats. Additionally, we incorporate various font styles, including novel designer fonts sourced from the web (Apple Inc., n.d.; Fonts.net.cn, n.d.).

4.3 Quantitative Results

Table 1 presents the quantitative results evaluated on the AnyText benchmark (Tuo et al., 2023), along with our proposed metrics $l_2@k$ and $\cos@k$ with $k = 5, 20, 50$, and the full logits, i.e., $c = 3474$ in the pretrained font classification model to assess font fidelity. Note that ControlText relies on a text segmentation model to generate glyph controls for the AnyText benchmark automatically. This may produce a small number of low-quality masks. However, we are not concerned about this, as human users can always type high-quality glyph controls during the actual use; therefore, those scores only serve as lower bounds. To ensure a fair comparison of all methods, we filter out text with low-quality masks based on the same criterion described in Section 3.2.1 before calculating all metrics.

While ControlText shows some differences compared to AnyText in SenACC and NED on Chinese characters, it successfully maintains large gaps across metrics on English data and fuzzy font accuracy. Meanwhile, when using identical font-aware glyph controls in ControlText, AnyText experiences a substantial decrease in text accuracy with almost no improvement in font accuracy, as shown in the row marked "AnyText-v1.1 Font Aware" in Table 1. This demonstrates ControlText's superior ability to handle diverse and nuanced font variations without requiring fine-tuning for each font.

5 Discussion

This work presents a simple and scalable proof-of-concept for multilingual visual text rendering with user-controllable fonts in the open world. We summarize our key findings as follows:

Font controls require no font label annotations A text segmentation model can capture nuanced font information in pixel space without requiring font label annotations in the dataset, enabling zero-shot generation on unseen languages and fonts, as well as scalable training on web-scale image datasets as long as they contain text.

Evaluating ambiguous fonts in the open world Fuzzy font accuracy can be measured in the embedding space of a pretrained font classification model, utilizing our proposed metrics $l_2@k$ and $\cos@k$.

Supporting user-driven design flexibility Random perturbations can be applied to segmented glyphs. While this won't affect the rendered text quality, it accounts for users not precisely aligning text to best locations and prevents models from rigidly replicating the pixel locations in glyphs.

Working with foundation models With limited computational resources, we can still copilot foundational image generation models to perform localized text and font editing.

Future work will focus on enhancing data efficiency, especially reinforcement pipeline and fonts in low-resource languages, as well as enabling more complicated artistic style control of text from user prompts beyond font information, including its interaction with diverse underlying background.

6 Limitations

Our model is based on ControlNet (Zhang et al., 2023b) with a CLIP text embedding model (Radford et al., 2021), although modified by AnyText (Tuo et al., 2023) to incorporate glyph line information. However, the CLIP-text encoder has relatively limited language understanding capabilities compared to state-of-the-art foundation models. Unlike text itself, this limitation affects the model's ability to accurately render complex artistic visual features or backgrounds, which users might specify in their input prompts, such as asking the text to appear like clouds or flames, that go beyond merely the font information.

Additionally, due to limited training resources, our experiments were conducted using a smaller diffusion model as a proof-of-concept compared to commercial ones. Each epoch requires approximately 380 GPU hours on NVIDIA V100 GPUs with 32 GB of memory, but we anticipate significantly improved efficiency on newer hardware and with a larger memory. This constraint may result in suboptimal inpainting of background regions within the text area, as well as instability in the quality of rendered text. The users also have limited controls of background pixels behind the text.

Some sacrifice in text quality is observed for non-Latin languages on the AnyText-Benchmark in exchange for improved font controllability.

The embedding layers of the glyph controls can also lead to reduced text quality, especially when the text in a font is very small, thin, or excessively long. In such cases, fine details of the font information in the glyphs may be lost.

As with all other text-to-image algorithms that rely on diffusion models, our approach requires a certain number of denoising steps to generate a single image at inference. End-to-end transformer-based models (Xie et al., 2024) may improve the time efficiency of the generation process.

7 Ethical Impact

This work is intended solely for academic research purposes. While our algorithm allows users to generate images with customized text, there is a potential risk of misuse for producing harmful or hateful content or misinformation. However, we do not identify any additional ethical concerns compared to existing research on visual text rendering.

References

Flux1 AI. Flux1 ai.

Storia AI. 2025. Font classify. Accessed: 2025-01-12.

Apple Inc. n.d. Apple fonts. https://developer.apple.com/fonts/. Accessed: 2025-02-01.

Yuhang Bai, Zichuan Huang, Wenshuo Gao, Shuai Yang, Jiaying Liu, et al. 2024. Intelligent artistic typography: A comprehensive review of artistic text design and generation. *APSIPA Transactions on Signal and Information Processing*, 13(1).

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. 2023. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8.

Fengxiang Bie, Yibo Yang, Zhongzhu Zhou, Adam Ghanem, Minjia Zhang, Zhewei Yao, Xiaoxia Wu, Connor Holmes, Pareesa Golnari, David A Clifton, et al. 2023. Renaissance: A survey into ai text-to-image generation in the era of large model. *arXiv preprint arXiv:2309.00810*.

Business Insider. 2011. 5 cheapo take-out options when you're too lazy to cook holiday dinner. https://www.businessinsider.com/5-cheapo-take-out-options-when-youre-too-lazy-to-cook-h Accessed: 2025-02-01.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. 2022. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325.

Haoxing Chen, Zhuoer Xu, Zhangxuan Gu, Yaohui Li, Changhua Meng, Huijia Zhu, Weiqiang Wang, et al. 2024a. Diffute: Universal text editing diffusion model. *Advances in Neural Information Processing Systems*, 36.

Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. 2024b. Textdiffuser: Diffusion models as text painters. *Advances in Neural Information Processing Systems*, 36.

Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. 2025. Textdiffuser-2: Unleashing the power of language models for text rendering. In *European Conference on Computer Vision*, pages 386–402. Springer.

Yejin Choi, Jiwan Chung, Sumin Shim, Giyeong Oh, and Youngjae Yu. 2024. Towards visual text design transfer across languages. *arXiv preprint arXiv:2410.18823*.

CNN Travel. n.d. Best classic restaurants in new york city. https://www.cnn.com/travel/article/best-classic-restaurants-new-york-city/index.html. Accessed: 2025-02-01.

DeepFloyd-Lab. 2023. Deepfloyd if. https://github.com/deep-floyd/IF.

Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, et al. 2020. Pp-ocr: A practical ultra lightweight ocr system. *arXiv preprint arXiv:2009.09941*.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*.

Fonts.net.cn. n.d. Chinese fonts collection. https://www.fonts.net.cn/fonts-zh-1.html. Accessed: 2025-02-01.

Zhen Han, Zeyinzi Jiang, Yulin Pan, Jingfeng Zhang, Chaojie Mao, Chenwei Xie, Yu Liu, and Jingren Zhou. 2024. Ace: All-round creator and editor following instructions via diffusion transformer. *arXiv preprint arXiv:2410.00086*.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Jiun Tian Hoe, Xudong Jiang, Chee Seng Chan, Yap-Peng Tan, and Weipeng Hu. 2024. Interactdiffusion: Interaction control in text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6180–6189.

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

Chao Li, Chen Jiang, Xiaolong Liu, Jun Zhao, and Guoxin Wang. 2024a. Joytype: A robust design for multilingual visual text creation. *arXiv preprint arXiv:2409.17524*.

Hua Li and Zhouhui Lian. 2024. Hfh-font: Few-shot chinese font synthesis with higher quality, faster speed, and higher resolution. *ACM Transactions on Graphics (TOG)*, 43(6):1–16.

Wenbo Li, Guohao Li, Zhibin Lan, Xue Xu, Wanru Zhuang, Jiachen Liu, Xinyan Xiao, and Jinsong Su. 2024b. Empowering backbone models for visual text generation with input granularity control and glyph-aware training. *arXiv preprint arXiv:2410.04439*.

Zhenhang Li, Yan Shu, Weichao Zeng, Dongbao Yang, and Yu Zhou. 2024c. First creating backgrounds then rendering texts: A new paradigm for visual text blending. *arXiv preprint arXiv:2410.10168*.

Rosanne Liu, Dan Garrette, Chitwan Saharia, William Chan, Adam Roberts, Sharan Narang, Irina Blok, RJ Mical, Mohammad Norouzi, and Noah Constant. 2022. Character-aware models improve visual text rendering. *arXiv preprint arXiv:2212.10562*.

Zeyu Liu, Weicong Liang, Zhanhao Liang, Chong Luo, Ji Li, Gao Huang, and Yuhui Yuan. 2025. Glyph-byt5: A customized text encoder for accurate visual text rendering. In *European Conference on Computer Vision*, pages 361–377. Springer.

Zeyu Liu, Weicong Liang, Yiming Zhao, Bohan Chen, Lin Liang, Lijuan Wang, Ji Li, and Yuhui Yuan. 2024. Glyph-byt5-v2: A strong aesthetic baseline for accurate multilingual visual text rendering. *arXiv preprint arXiv:2406.10208*.

Jian Ma, Yonglin Deng, Chen Chen, Haonan Lu, and Zhenyu Yang. 2024. Glyphdraw2: Automatic generation of complex glyph posters with diffusion models and large language models. *arXiv preprint arXiv:2407.02252*.

Jian Ma, Mingjun Zhao, Chen Chen, Ruichen Wang, Di Niu, Haonan Lu, and Xiaodong Lin. 2023. Glyphdraw: Seamlessly rendering text with intricate spatial structures in text-to-image generation. *arXiv preprint arXiv:2303.17870*.

Midjourney. Midjourney.

AI Monks. 2023. 30 stunning realistic midjourney prompts you can use. https://medium.com/aimonks/30-stunning-realistic-midjourney-prompts-you-can-use-52 Accessed: 2025-02-01.

Nipic. n.d. Image resource from nipic. https://nipic.com/show/42990265.html. Accessed: 2025-02-01.

Shubham Paliwal, Arushi Jain, Monika Sharma, Vikram Jamwal, and Lovekesh Vig. 2024a. Customtext: Customized textual image generation using diffusion models. *arXiv preprint arXiv:2405.12531*.

Shubham Singh Paliwal, Arushi Jain, Monika Sharma, Vikram Jamwal, and Lovekesh Vig. 2024b. Orientext: Surface oriented textual image generation. In *SIGGRAPH Asia 2024 Technical Communications*, pages 1–4.

peterpom211. 2024. Post on x (formerly twitter). https://x.com/peterpom211/status/1871890976069091377?mx=2. Accessed: 2025-02-01.

PIXTA. n.d. Stock photo from pixta (id: 93233725). https://www.pixtastock.com/photo/93233725. Accessed: 2025-02-01.

Zhipeng Qian, Pei Zhang, Baosong Yang, Kai Fan, Yiwei Ma, Derek F Wong, Xiaoshuai Sun, and Rongrong Ji. 2024. Anytrans: Translate anytext in the image with large scale models. *arXiv preprint arXiv:2406.11432*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr.

Anton Razzhigaev, Arseniy Shakhmatov, Anastasia Maltseva, Vladimir Arkhipkin, Igor Pavlov, Ilya Ryabov, Angelina Kuts, Alexander Panchenko, Andrey Kuznetsov, and Denis Dimitrov. 2023. Kandinsky: an improved text-to-image synthesis with image prior and latent diffusion. *arXiv preprint arXiv:2310.03502*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494.

Jaejung Seol, Seojun Kim, and Jaejun Yoo. 2025. Posterllama: Bridging design ability of language model to content-aware layout generation. In *European Conference on Computer Vision*, pages 451–468. Springer.

Wenda Shi, Yiren Song, Dengming Zhang, Jiaming Liu, and Xingxing Zou. 2024. Fonts: Text rendering with typography and style controls. *arXiv preprint arXiv:2412.00136*.

Yanan Sun, Yanchen Liu, Yinhao Tang, Wenjie Pei, and Kai Chen. 2025. Anycontrol: create your artwork with versatile control on text-to-image generation. In *European Conference on Computer Vision*, pages 92–109. Springer.

Richard Sutton. 2019. The bitter lesson. *Incomplete Ideas (blog)*, 13(1):38.

Tripadvisor. n.d. Chicago's dark side self-guided smartphone walking tour. https://www.tripadvisor.com/AttractionProductReview-g35805-d25240549-Chicago_s_Dark_Side_Self_Guided_Smartphone_Walking_Tour-Chicago_Illinois.html. Accessed: 2025-02-01.

Yuxiang Tuo, Yifeng Geng, and Liefeng Bo. 2024. Anytext2: Visual text generation and editing with customizable attributes. *arXiv preprint arXiv:2411.15245*.

Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Geng, and Xuansong Xie. 2023. Anytext: Multilingual visual text generation and editing. *arXiv preprint arXiv:2311.03054*.

Unsplash. n.d. Unsplash: Free high-resolution photos. https://unsplash.com. Accessed: 2025-02-01.

Aoqiang Wang, Jian Wang, Zhenyu Yan, Wenxiang Shang, Ran Lin, and Zhao Zhang. 2024a. Textmaster: Universal controllable text edit. *arXiv preprint arXiv:2410.09879*.

Yibin Wang, Weizhong Zhang, Changhai Zhou, and Cheng Jin. 2024b. High fidelity scene text synthesis. *arXiv preprint arXiv:2405.14701*.

Wikipedia contributors. n.d. Théâtre d'opéra spatial. https://en.wikipedia.org/wiki/Th%C3%A9%C3%A2tre_D%27op%C3%A9ra_Spatial. Accessed: 2025-02-01.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. 2024. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*.

Xingqian Xu, Zhifei Zhang, Zhaowen Wang, Brian Price, Zhonghao Wang, and Humphrey Shi. 2021. 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*, pages 12045–12055.

Ling Yang, Zhilong Zhang, Zhaochen Yu, Jingwei Liu, Minkai Xu, Stefano Ermon, and CUI Bin. 2024a. Cross-modal contextualized diffusion models for text-guided visual generation and editing. In *The Twelfth International Conference on Learning Representations*.

Yukang Yang, Dongnan Gui, Yuhui Yuan, Weicong Liang, Haisong Ding, Han Hu, and Kai Chen. 2024b. Glyphcontrol: Glyph conditional control for visual text generation. *Advances in Neural Information Processing Systems*, 36.

Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, and In So Kweon. 2023a. Text-to-image diffusion models in generative ai: A survey. *arXiv preprint arXiv:2303.07909*.

Lingjun Zhang, Xinyuan Chen, Yaohui Wang, Yue Lu, and Yu Qiao. 2024. Brush your text: Synthesize any scene text on images via diffusion model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 7215–7223.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023b. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847.

Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. 2023. Unleashing text-to-image diffusion models for visual perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5729–5739.

Xiaoran Zhao, Tianhao Wu, Yu Lai, Zhiliang Tian, Zhen Huang, Yahui Liu, Zejiang He, and Dongsheng Li. 2024a. Ltos: Layout-controllable text-object synthesis via adaptive cross-attention fusions. *arXiv preprint arXiv:2404.13579*.

Zhen Zhao, Jingqun Tang, Binghong Wu, Chunhui Lin, Shu Wei, Hao Liu, Xin Tan, Zhizhong Zhang, Can Huang, and Yuan Xie. 2024b. Harmonizing visual text comprehension and generation. *arXiv preprint arXiv:2407.16364*.

Yuanzhi Zhu, Jiawei Liu, Feiyu Gao, Wenyu Liu, Xinggang Wang, Peng Wang, Fei Huang, Cong Yao, and Zhibo Yang. 2024. Visual text generation in the wild. *arXiv preprint arXiv:2407.14138*.