Beyond Flat Text: Dual Self-inherited Guidance for Visual Text Generation

Minxing Luo[1]*, Zixun Xia[1]*, Liaojun Chen[1], Zhenhang Li[2], Weichao Zeng[2], Jianye Wang[1], Wentao Cheng[1], Yaxing Wang[1]†, Yu Zhou[1] and Jian Yang[1]†

[1] VCIP, CS, Nankai University, [2] Institute of Information Engineering, Chinese Academy of Sciences

A signpost stands at the junction of garden paths, with the words "Paris" "Tokyo" and "Nashville" displayed on the sign.

A romantic dinner table set for two, with candles and flowers arranged in a V shape, displaying "Forever" and "Love" .

A vibrant street art mural in a lively city square, showcasing "ICCV" in bold, artistic graffiti surrounded bright splashes of color.

A cheerfully glowing pumpkin lantern with the bright and shining text "Trick" "Or" "Treat" is intricately carved at the center, creating a warm atmosphere.

A vibrant forest with a playful unicorn frolicking among trees and the words "Mystic Realm" floating in the air.

Figure 1. **STGen for visual text generation in challenging layout.** *Using a pre-trained visual text generation model (e.g., AnyText [33]), our method, STGen, guides the model to adjust the text region in latent space during image synthesis, producing images that more faithfully represent the input prompt with precise visual text.*

Abstract

In real-world images, slanted or curved texts, especially those on cans, banners, or badges, appear as frequently, if not more so, than flat texts due to artistic design or layout constraints. While high-quality visual text generation has become available with the advanced generative capabilities of diffusion models, these models often produce distorted text and inharmonious text backgrounds when given slanted or curved text layouts due to training data limitations. In this paper, we propose a new framework, STGen, which accurately generates visual texts in challenging scenarios (e.g., slanted or curved text layouts) while harmonizing them with the text background. Our framework decomposes the visual text generation process into two branches: (i) ***Semantic Rectification Branch****, which leverages the ability in generating flat but accurate visual texts of the model to guide the generation of challenging scenarios. The generated latent of flat text is abundant in accurate semantic information related to both the text itself and its background. By incorporating this, we rectify the semantic information of the texts and harmonize the integration of the text with its background in complex layouts. (ii)* ***Structure Injection Branch****, which reinforces the visual text structure during inference. We incorporate the latent information of the glyph image, rich in glyph structure, as a new condition to further strengthen the text structure. To enhance image harmony, we also apply an effective combination method to merge the priors, providing a solid foundation for generation. Extensive experiments across a variety of visual text layouts demonstrate that our framework achieves superior accuracy and outstanding quality.*

1. Introduction

Visual text generation is an emerging yet challenging research area in image generation because text is fine-grained and difficult to balance with the image. The current methods [1, 3, 5, 6, 14, 16, 17, 19, 23, 29, 32–35, 40] proposed a series techniques to address it. Among them, diffusion based methods (AnyText [33], GlyphControl [35], DiffText [38] *etc.*) can create new images with integrated text, raising an inevitable challenge to achieve both accurate visual texts and harmonious image content. Notably, AnyText [33] distinguishes itself by producing impressive images integrated with outstanding multilingual text. It opens an era of universal visual text generation using large pretrained Visual Text Generation Model (VTGM).

Although large pretrained VGTMs take a significant step towards universal visual text generation, they still struggle to handle users' diverse inputs, such as commonly seen

* Equal contribution.
† Corresponding author.

Figure 2. **Failure cases of AnyText [33].** *The top row illustrates two failure cases: textual distortion (left) and background occlusion (right). The bottom row displays results using our method.*

slanted or curved texts in real-world images. Given these user inputs, the model often leads to text distortion and background occlusion, as shown in Fig. 2. This is because the latent space gradually becomes blurry and distorted (as shown in Fig. 3) due to insufficient data in such scenarios, thus the model cannot effectively maintain the structure and semantic information in the text region as it does when processing a flat mask.

One naive solution is to train VTGM on a more diversified dataset that covers various text configurations. But it is resource-intensive and data distribution across multiple text layouts cannot be guaranteed. Even if they are trained on those data, the results may not be satisfying [9, 30].

To overcome this shortcoming, we propose a plug-and-play method named Slanted Text Generation (STGen), which corrects text regions in latent space during inference. Specifically, our approach employs a dual-branch framework. The first branch, the *Semantic Rectification Branch (SRB)*, utilizes a latent generated using the same prompt, but with a simplified shape, as a robust semantic prior. This branch simultaneously rectifies distorted text predictions and harmonizes the text with its background. For complex visual text generation, such as text layouts composed of multiple tilted or curved sections, we propose a *Divide and Conquer* strategy to efficiently reconfigure the text shape and obtain a reasonable semantic prior.

The second branch, *Structure Injection Branch (SIB)*, extracts rich structural information from glyphs and injects it into the latent space as a structural prior, further enhancing the accuracy of the visual text. Rather than simply merging the two priors, we adopt a novel combination method for optimized integration for better coherence in latent space. Together, the dual-branch framework offers effective guidance for the generation process without requiring additional training.

To the best of our knowledge, our method is the first tailored specifically for generating visual texts in complex layouts and achieves state-of-the-art results as shown in

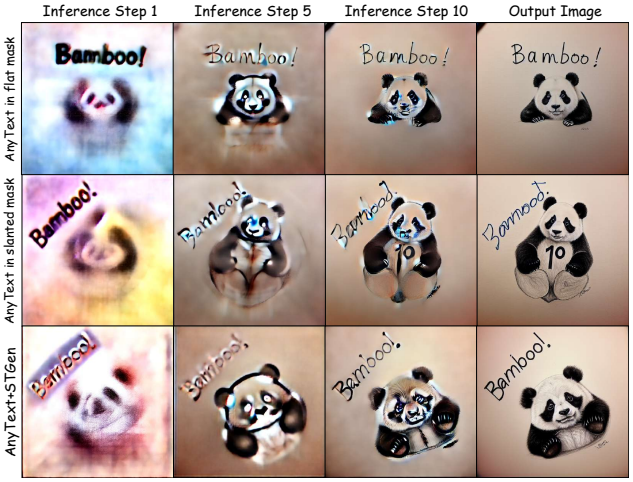

Figure 3. **Comparison of predicted x_0 under different inference steps.** *The first and second rows show AnyText's intermediate predicted x_0 for flat and slanted masks, respectively. While predictions remain stable under flat masks, the x_0 prediction drifts during inference with slanted masks. Our method effectively guides the model to maintain accuracy in visual texts.*

Figs. 1, 5 and 6 without additional training. Our framework effectively improves visual text accuracy across various approaches, including early-stage GlyphControl, DiffText, which receives no specified training for visual text generation, and AnyText. We conduct thorough experiments to demonstrate the superiority of our method and the effectiveness of each component in generating complex visual texts while enhancing overall image quality and coherence. The main contributions of our work are summarized as follows:

(i) We present a new challenge: generating visual text in complex layouts with diffusion models. To tackle this, we introduce STGen, a dual-branch, training-free framework that can be easily integrated with existing text generation models. This framework enables the model to generate text in complex layouts by incorporating both structural and semantic priors.

(ii) We introduce an effective method to combine given priors within the latent space. This approach seamlessly merges priors and the latent, ensuring a consistent latent range and resulting in a balanced, high-quality image.

(iii) We propose a benchmark extending the AnyText-benchmark to evaluate texts in complex layouts, enabling comprehensive assessment across varying text slant difficulties.

2. Related Works

Denoising diffusion probabilistic models (DDPMs) [4, 7, 11, 21, 26–29, 31] have made significant contribution to the

field of text-to-image synthesis. However, achieving precise control over image details remains challenging, particularly in visual text generation.

2.1. Local Visual Text Blending

Local Visual Text Blending methods synthesize localized images with single-line textual content. Existing approaches typically necessitate a background image for text insertion or modification. SynText [8] identifies image regions suitable for text placement and renders text accordingly. Following their work, SceneVTG [40] introduces a model to predict the text-generating area and specifically designs a framework that takes line masks and word masks separately to help generate texts in different shapes. SceneVTG is less user-friendly because it requires an erased image and lacks support for multilingual text generation. AnyTrans [22] detects, removes, and replaces text with translations in the same region. The above methods require an input image to render or generate, which is good for editing, but cannot generate the whole image from scratch.

2.2. Global Visual Text Generation

Recent studies, such as Imagen [29], demonstrate that replacing CLIP text encoder [24] with more advanced models like T5 [25] enhances visual text generation. Liu *et al*. [16] further replace the character-blind text encoder with a character-aware text encoder. GlyphByT5 series [17, 18] employ character-aware ByT5 encoder [15] and a new cross attention mechanism to compute the text region and image region separately. Although replacing text encoders appears straightforward, it still struggles to generate complex characters such as Korean, Japanese, and Chinese. GlyphDraw [19] pioneered using glyph conditions and location masks for visual text generation of complex characters. It employs a glyph image and a location mask to control content and placement, though it remains limited to one line per image. TextDiffuser [6] introduces a dedicated layout generation module to produce character-level masks as diffusion model conditions, enhancing Latin text generation. But it still cannot generate non-Latin texts. TextDiffuser-2 [5] refines this approach by replacing character-level masks with bounding box coordinates, but this modification weakens its capacity for flexible visual text layout customization. GlyphControl [35] leverages ControlNet [37] and uses rendered glyph images as control conditions, enabling visual text generation in models like Stable Diffusion while preserving their core image synthesis capabilities. Nevertheless, GlyphControl is restricted to straight-line text layouts and frequently generates extraneous text artifacts. Building on these works, AnyText [33] proposes a unified framework for high-quality multilingual visual text generation. Its text perceptual loss directly evaluates text accuracy during training, improving correctness of generated visual text. However, AnyText struggles with rendering text rotated beyond 45 degrees. TextGen [36] examines how control signals affect image generation across timesteps. TextHarmony [39] unifies image comprehension, generation, and editing within a single framework by training a multimodal Large Language Model (LLM) to improve image understanding and generate more precise tokens for image synthesis. However, users face challenges in defining text layout during image generation, as current methods rely solely on textual prompts. In terms of text editing, its performance in complex backgrounds is unsatisfactory. Diff-Text leverages Canny ControlNet for glyph-based generation but ties text placement to predefined objects in prompts. It may fail if a prompt does not include predefined objects, leading to unnatural images.

3. Method

Given a prompt y describing the image and a challenging position mask l_p, our goal is to generate an image based on y that incorporates visual text at the specified positions dictated by l_p. Current visual text generation models struggle to generate visual texts in more challenging l_p settings, such as tilted or curved text layouts.

Our method aims to solve the problem based on two key insights: First, when the position mask l_p is flat, current models can generate visual texts with high accuracy. We leverage this proficiency with flat visual texts to address challenging scenarios, providing a strong semantic prior for generation. Second, the glyph image contains little semantic information but is rich in glyph structural details. It can serve as a structure prior to further refine the structural information of the visual texts within the latent, thus enabling it to generate accurate slanted visual texts.

Our approach consists of two main branches: the *Semantic Rectification Branch* and the *Structure Injection Branch*. As shown in Fig. 4, the semantic rectification branch first takes the masks $\bar{l}_p$ reconfigured from l_p, prompt y, and random noise z_T to generate the latent representation z_0^f, which contains reconfigured and flat visual text as a semantic prior. The structure injection branch then takes the rendered glyph image l_g and z_t to generate a structural prior. These two priors are merged and fed into the VTGM, along with y, l_p, and l_g, for the denoising process. Further details of these branches are discussed in Sec. 3.1 and Sec. 3.2.

3.1. Semantic Rectification Branch

In the initial stages of the DDIM denoising process in the VGTM, clear text is generated. This text remains accurate when a flat position mask is applied. However, when using a slanted mask, the text gradually becomes distorted. The reason for the distortion lies in the semantic drift in the text region when tilted. Motivated by this observation, we inherit the high-quality flat text generation capabilities of the

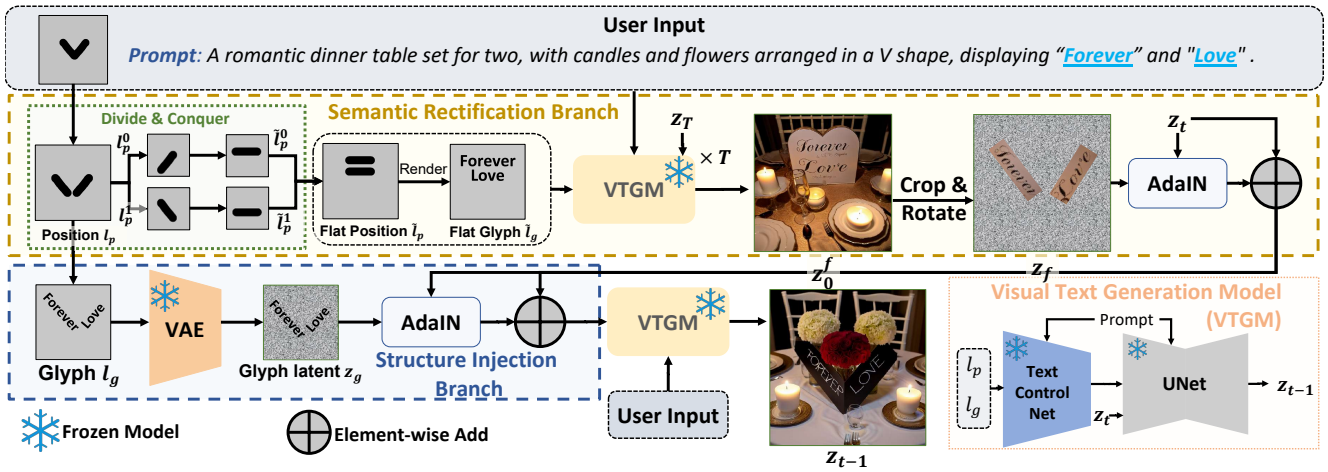

Figure 4. **Pipeline of our method**. *Given user input at the top, which contains a prompt and a mask l_p specifying positions for generating visual texts, we first split the l_p using Divide and Conquer Strategy and obtain glyph image l_g and flat position mask $\tilde{l}_p$. Then $\tilde{l}_p$ and l_g are input to the Semantic Rectification Branch and Structure Injection Branch respectively. In the Semantic Rectification Branch, based on $\tilde{l}_p$, we render flat glyph $\tilde{l}_g$ and use them along with prompt and random noise z_T to generate the latent with flat visual text. This latent serves as a semantic prior, providing rich semantic information for both the generation of the text and its background. l_g, on the other hand, is converted into the latent space as a structural prior for structural refinement of the text. Finally, the two prior combined to guide the generation of the visual text in l_p.*

existing visual text model, adopting the latent of flat text as a constant reference to rectify the semantic information for challenging text generation.

Reference Branch for Semantic Rectification. Parallel to the generation branch, a separate branch is employed to generate flat visual texts using the same prompt y for additional semantic information, which we refer to as the semantic rectification branch. In this branch, the flat visual text latent is blended into the tilted one within the text region, while the rest of the latent remains unchanged.

As shown in Fig. 4, the flat position $\tilde{l}_p$ and corresponding glyph image $\tilde{l}_g$ are fed into the VTGM along with prompt y and random noise z_T. After T denoising steps, we obtain the reference latent z_0^f. We rotate z_0^f to match the user-given position, and extract its text region z_f as a semantic guide for the branch below:

$$\tilde{z}_t = z_f \odot l_p + z_t \odot (1 - l_p), \tag{1}$$

where l_p is the position mask input by the user. This operation effectively rectifies the visual text using the accurate text in the z_f. Thanks to the faithful background semantic information embedded in the z_f, we significantly reduce the erroneous non-textual semantic information in the text region while maintaining a coherent background, thereby avoiding background occlusion.

AdaIN Combination. AdaIN [12] is originally developed for style transfer tasks. It substitutes mean and standard deviation of the source feature with those of the target feature. Masui *et al*. [20] demonstrated that AdaIN can be applied directly to Diffusion models for style transfer without any additional training. In our method, we also incorporate AdaIN, but to preserve image integrity against disruptions in latent distribution caused by the replacement operation:

$$\begin{aligned}\tilde{z}_t &= \mathrm{AdaIN}(z_f, z_t) \odot l_p + z_t \odot (1 - l_p) \\ &= (\sigma(z_t)(\frac{x - \mu(z_f)}{\sigma(z_f)}) + \mu(z_t)) \odot l_p + z_t \odot (1 - l_p),\end{aligned} \tag{2}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ denote the channel-wise mean and standard deviation. The aid of AdaIN ensures the latent for guidance has the same range as the original latent, minimizing the change in latent distribution. With a more consistent latent, the visual text semantic information is further enhanced while preserving the overall image coherence.

Divide and Conquer Strategy. In practice, complex position masks are often provided for generating realistic visual texts, which often consist of multiple straight or curved parts, bringing obstacles to the geometric alignment to flat position masks $\tilde{l}_p$ in our method. To deal with the problem, we propose a divide-and-conquer strategy to effectively segment complex position masks into more manageable straight sections as shown in Fig. 4. We use Bézier curves to define the upper and lower boundaries of l_p and establish a baseline for the texts by averaging these curves. We identify splitting points on the curve where the direction vectors at each point are parallel to the boundaries of the minimum bounding box of l_p, therefore creating N re-

configured masks:

$$l_p = \left\{ l_p^0, l_p^1, \ldots, l_p^{N-1} \right\}. \tag{3}$$

Subsequently, we rotate and regroup these reconfigured masks to obtain flat position masks $\tilde{l}_p$ and render corresponding glyphs $\tilde{l}_g$:

$$\tilde{l}_p = \left\{ \tilde{l}_p^0, \tilde{l}_p^1, \ldots, \tilde{l}_p^{N-1} \right\}, \tilde{l}_g = \left\{ \tilde{l}_g^0, \tilde{l}_g^1, \ldots, \tilde{l}_g^{N-1} \right\}. \tag{4}$$

With the assistance of this branch, we can generate highly accurate texts and create harmonious, diverse images by leveraging rich semantic information in both the visual texts and background.

3.2. Structure Injection Branch

Despite semantic guidance, the predictions may still deviate due to limited information in structure, causing cumulative errors and structural inconsistencies that require enriched structural guidance for clarity. To resolve this, we introduce a glyph structure prior directly into the latent space. Unlike early methods [19] that use glyph images directly as auxiliary information, we consider text glyphs as essential parts of the image, which should be comprehended by the visual text generation model in the latent space. As shown in Fig. 4, the structure injection branch feeds the rendered glyph image l_g to Variational Autoencoder (VAE) [13] encoder to obtain the latent z_g. As an extension of semantic prior, we incorporate the z_g as a structure prior into the latent z_t, using the same AdaIN operation above to regulate the range, which is represented as:

$$\hat{z}_t = \text{AdaIN}(z_g, z_t) \odot l_p + z_t \odot (1 - l_p). \tag{5}$$

This operation provides the model with a structural foundation that serves as a strong starting point in the generation process, further enhancing the overall structure of the visual texts.

Combining the two branches, the merged prior is represented by:

$$\hat{z}_t = \rho \text{AdaIN}(z_g, z_t) + (1 - \rho)\tilde{z}_t, \tag{6}$$

where ρ is a hyper-parameter that adjusts the balance between semantic and structural information. The modified latent is represented as follows:

$$\ddot{z}_t = (\kappa_t \lambda \hat{z}_t + (1 - \kappa_t) z_t) \odot l_p + z_t \odot (1 - l_p), \tag{7}$$

where λ is a hyper-parameter and the κ_t is a temporal factor that decays over time with each timestep. Together, they control the injection strength of the merged prior.

Language	λ	ρ	Sen.Acc↑	NED↑	CLIP Score↑
English	-0.5	0.5	44.88	64.11	0.3005
	0.5	0.5	**45.43**	65.10	0.3027
	0.5	0.25	45.12	63.85	**0.3036**
	0.5	0.75	45.10	**65.65**	0.3006
	0.5	1.50	45.01	65.34	0.3009
	0.5	2.00	44.53	65.17	0.3009
Chinese	-0.5	0.5	49.28	87.29	0.3067
	0.5	0.5	49.96	**88.08**	0.3071
	0.5	0.25	**50.70**	87.86	**0.3076**
	0.5	0.75	49.05	87.89	0.3058
	0.5	1.50	47.40	87.64	0.3061
	0.5	2.00	47.88	87.63	0.3059

Table 1. **Sensitivity analysis for** $[\lambda, \rho]$**.**

4. Experiments

4.1. Implementation Details

We use a single RTX 3090 GPU for images of size (512, 512). Optimal performance is achieved when $\lambda \in [-0.5, 0.5]$ and $\rho \in (0, 2]$. We set $\kappa_t = 10^{t-T}$ for effective guidance and harmonious text region boundary.As shown in Tab. 1, our method demonstrates robustness for hyper-parameter sensitivity. In subsequent experiments, we set λ to 0.5 and ρ to 0.5, which has a balanced performance on both Chinese and English for evaluation.

4.2. Evaluation Setup

Due to the lack of publicly available datasets focused on challenging visual text generation, we propose a new benchmark derived from the AnyText-benchmark [33]. For each text position mask, we randomly rotate the original benchmark's masks and resolve overlaps, yielding 984 prompts for LAION-word (English evaluation) and 919 prompts for Wukong-word (Chinese evaluation). During evaluation, masks are categorized into three difficulty levels based on rotation angles: easy (0°–30°), medium (30°–60°), and hard (60°–90°), enabling multi-level assessment of performance.

Textual accuracy and background-text coherence are two main factors that determine the quality of slanted text generation, which we quantitatively evaluate through OCR accuracy. Following AnyText [33], we select the following two metrics for comparing OCR accuracy at word-level and character-level, respectively: (1) Sentence Accuracy (Sen.Acc); (2) Normalized Edit Distance (NED).

We evaluated existing competing methods, including TextDiffuser [6], TextDiffuser-2 [5], TextHarmony [39], GlyphControl [35], SceneVTG [40], AnyText [33] and Diff-Text [38] using the benchmark and metrics mentioned above. Notably, SceneVTG [40] cannot generate images from scratch. To address this, we use background images produced by Stable Diffusion [28] with identical prompts. For TextHarmony [39], its image generation model lacks support for positioning visual text at specific locations. Following its evaluation protocol on the AnyText Benchmark,

Language	Methods	Sen.Acc↑				NED↑				CLIP Score↑
		easy	medium	hard	total	easy	medium	hard	total	
English	TextDiffuser [6]	49.88	20.10	0.262	29.06	70.86	41.25	4.47	45.78	0.3091
	TextDiffuser-2 [5]	0.59	0.00	0.15	0.32	3.23	0.60	0.50	1.81	0.2989
	SD1.5+TextHarmony† [39]	1.14	0.00	0.15	0.58	16.85	2.51	1.12	8.92	0.3090
	GlyphControl [35]	19.00	1.63	0.22	9.47	41.74	9.17	2.98	22.95	**0.3206**
	SD1.5+SceneVTG† [40]	13.62	4.97	1.12	8.06	24.14	13.40	3.00	15.80	0.3112
	Anytext [33]	62.77	29.12	2.02	38.04	83.64	56.99	14.23	58.59	0.3007
	Diff-Text [38]	40.13	26.42	8.85	28.38	61.44	45.37	17.94	45.91	0.2962
	Diff-Text+Ours	62.23	**50.99**	**27.28**	**50.21**	76.49	64.34	**37.49**	63.18	0.3018
	GlyphControl+Ours	39.19	2.63	1.05	19.48	55.57	11.83	6.74	31.17	0.3175
	AnyText+Ours	**71.25**	37.25	6.60	45.43	**87.54**	**64.42**	24.56	**65.10**	0.3027
Chinese	TextDiffuser [6]	5.41	4.07	0.09	4.07	56.81	49.37	43.01	52.47	0.3066
	TextDiffuser-2 [5]	0.06	0.08	0.00	0.05	7.28	4.81	6.04	6.49	0.2984
	SD1.5+TextHarmony† [39]	0.00	0.00	0.00	0.00	14.35	8.50	12.00	12.60	0.3104
	GlyphControl [35]	2.33	0.24	0.0	1.41	50.98	15.58	35.94	40.24	**0.3192**
	SD1.5+SceneVTG† [40]	1.60	0.81	0.18	1.14	7.33	4.62	4.61	6.20	0.3025
	Anytext [33]	66.00	26.38	2.02	44.78	94.48	81.37	59.38	84.74	0.3064
	Diff-Text [38]	23.46	17.26	7.35	18.95	68.27	62.71	48.65	63.21	0.2950
	Diff-Text+Ours	42.13	**38.43**	**22.42**	37.47	81.15	71.84	59.69	74.91	0.2978
	GlyphControl+Ours	6.27	0.57	0.74	3.93	62.50	18.08	45.58	49.40	0.3168
	AnyText+Ours	**69.22**	35.58	8.55	**49.96**	**95.37**	**85.85**	**68.77**	**88.08**	0.3071

Table 2. **Quantitative Comparison between STGen and other competitors on both English and Chinese sets.** *All competitors are evaluated based on their officially released code and models. Numbers in **bold** indicate the best performance, and underscored numbers indicate the second best.* † *indicates we adapt the method which is originally focused on other tasks into image generation.*

Method	SEN.ACC ↑	NED ↑	FID ↓
GlyphControl	37.10/3.27	66.80/8.45	37.84/34.36
w/ STGen	**70.92/11.81**	**84.68/24.72**	**32.00/32.99**
Diff-Text	56.11/29.49	75.28/48.67	69.24/62.61
w/ STGen	**68.55/39.58**	**81.55/55.61**	**66.77/61.59**
AnyText	72.39/69.23	87.67/83.96	**33.54**/31.58
w/ STGen	**75.53/70.00**	**89.38/84.57**	33.83/**31.17**

Table 3. **Evaluation on vanilla AnyText Benchmark.** *In each cell of the table, the numbers on the right are results from the English set while the numbers on the right are from the Chinese set.*

Baseline	Baseline Preference	Ours Preference
TextDiffuser [6]	23.07%	76.93%
TextDiffuser-2 [5]	20.15%	79.85%
GlyphControl [35]	21.96%	78.04%
SceneVTG [40]	18.99%	81.01%
TextHarmony [39]	18.75%	81.25%
AnyText [33]	25.31%	74.69%
Diff-Text [38]	17.00%	83.00%

Table 4. **User study results.** *Participants were asked to choose the best results based on image quality, accuracy of generated text within images, and prompt-image similarity.*

we employ its visual text editing mode, which generates text on images where target positions are masked in black. For all the baselines in the evaluation, we use their officially released code and checkpoints. To ensure fairness, we preprocess the masks using the *Divide and Conquer* strategy mentioned above before generating images.

4.3. Quantitative Analysis

OCR Accuracy. As shown in Tab. 2, our method shows substantial improvements over the competitive methods in both English and Chinese across all levels. Notably, our method doesn't require additional training and enhances the performance of Diff-Text, GlyphControl, and AnyText at the easy level. At the hard level, AnyText with our method yields results closest to the ground truth in the NED metric, with an improvement of approximately 10%. over the baseline AnyText in both languages. As similarly shown in Tab. 3, our methods outperform baselines on vanilla AnyText Benchmark except FID. Meanwhile, Diff-Text with our method shows an impressive 20% improvement at the hard level in the English set.

Text-Image Similarities. In the absence of ground-truth images, we use the CLIP score [10] to assess the consistency between the prompt and the generated image. We compute the average cosine similarity between the prompt and the generated image, excluding the influence of the visual texts. As shown in Tab. 2, our method increases the

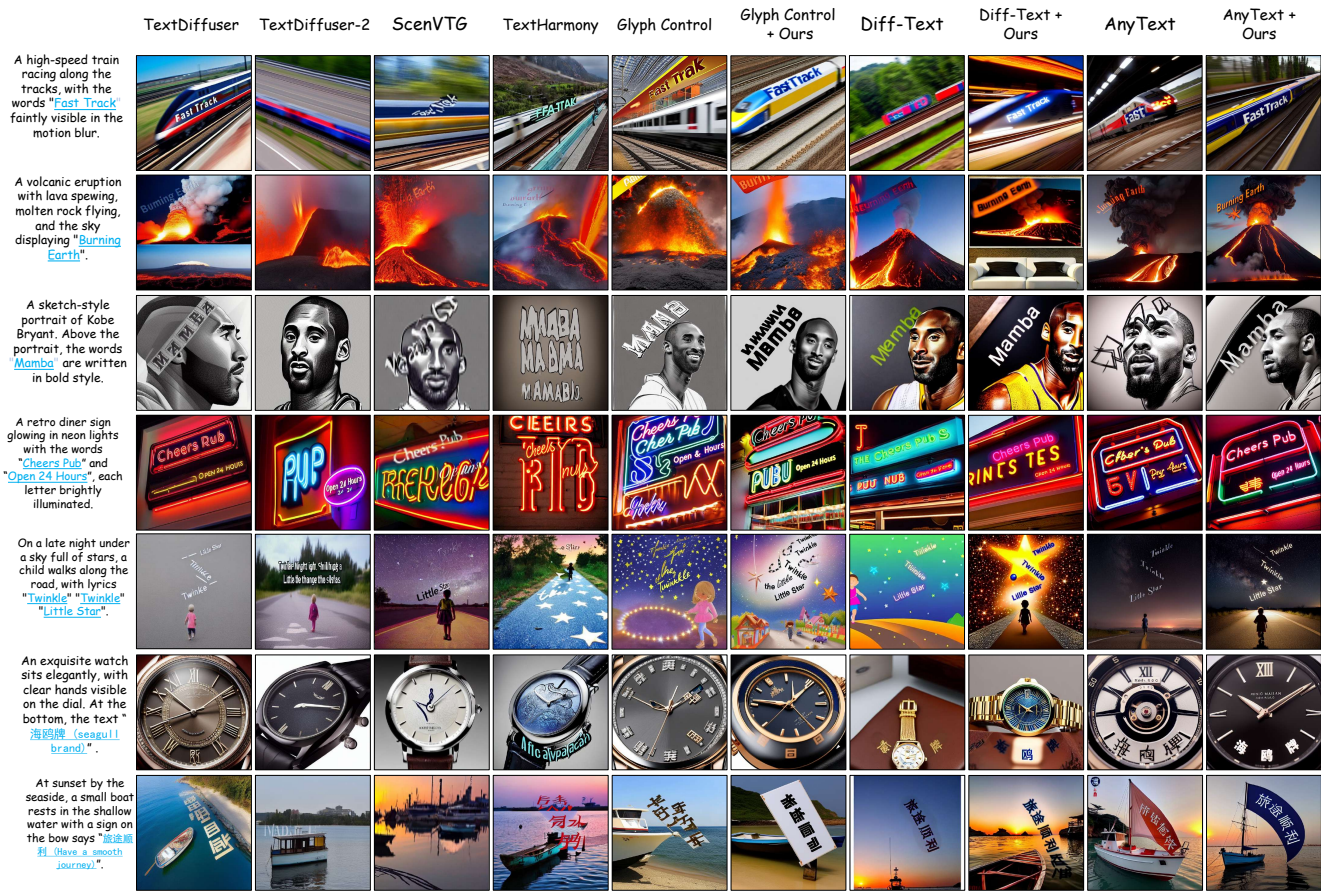

Figure 5. **Qualitative comparison of our method and state-of-the-art models in both English and Chinese text generation.**

accuracy of visual text generation without compromising the baseline performance. Additionally, our method slightly improves text-image similarity due to our design, which minimizes conflicts between the image composition and the visual text. This leads to a more coherent layout for both the image and the text.

User Study. Following [2], we conduct a user study to comprehensively compare the generation results in Tab. 4. We create 27 input sets of varying difficulty. For each set, participants receive our input prompt, position mask, and two images (our result and a baseline) in random order, and select the image with the best quality and most accurate visual text. The final score is the average number of selections per prompt. We gather 168 judgments from a diverse group of experts and non-experts and report vote percentages. As shown in Tab. 4, our method is preferred in all cases.

4.4. Qualitative Comparisons

As presented in Fig. 5, TextDiffuser [6] tends to generate low-quality images, specifically in the second and the fifth rows. Textdiffuser-2 [5] fails to consistently generate legible text within specified bounding boxes. We suspect this occurs due to interference between textual and non-textual elements when using overly large bounding boxes. GlyphControl [35] frequently produces texts outside its designated area, particularly when the text is near the image boundary, as shown in the second and fourth row. SceneVTG [40] suffers from severe text distortion, especially in the first and third rows. Background occlusion is also evident in the second, third, and fourth rows, where the texts "Burning," "Mamba," and "Twinkle" are obscured by lava, a human figure, and stars, respectively. This suggests SceneVTG struggles with complex backgrounds and lacks seamless integration with other text-to-image models. TextHarmony [39] faces similar issues. In image editing mode, it struggles to properly fill blanks with complex shapes, leading to distorted text, as seen in the second row. For Difftext [38], there is a noticeable lack of coherence in the images due to insufficient consideration for harmony between the visual texts and the background. AnyText [33] tends to produce distorted visual texts due to a loss of semantic and structural information during inference. For example, in the first row, the text "Track" overlaps with the train head, compromising both elements. In contrast, our method generates high-quality images with accurately rendered English

Figure 6. **More results on Background coherence and complex text layouts.** *The left two columns show how our approach seamlessly blends text with the background, while the remaining images highlight its ability to produce multi-sentence and circular text.*

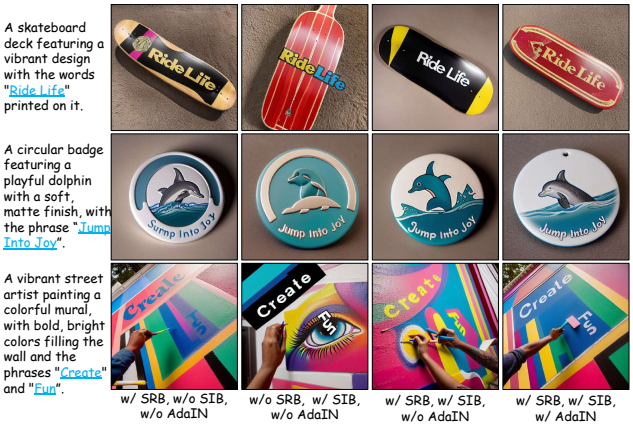

Figure 7. **Ablation Study Visualization.** *The first column shows accuracy improvement with SRB. The second reveals that SIB enhances accuracy but compromises coherence. The third confirms that combining both branches enhances both. Finally, the fourth shows that adding AdaIN further improves image quality.*

and Chinese texts—even in challenging scenarios such as those in the fourth and fifth rows and Fig. 6. Moreover, our approach not only corrects text distortion but also reduces background occlusion between textual and non-textual elements. For instance, in the third row of Fig. 5, AnyText [33] disrupts the text "Mamba" with a human head, whereas our method repositions the head to produce clear text alongside well-integrated non-text elements.

5. Ablation Study

In this section, we demonstrate the effectiveness of our components through an ablation study on the medium level. We use OCR accuracy (Sen.Acc and NED) and CLIP score as the main metrics, given their importance in evaluating visual text generation. All parameters are kept consistent with those outlined in Sec. 4.1.

SRB without AdaIN. As shown in the first two rows of Tab. 5, the addition of *Semantic Rectification Branch* improves the visual text accuracy. Similarly, as demonstrated by the first column of Fig. 7, the generated visual text has a clear structure. However, this also negatively impacts text-image consistency, as indicated by the slight drop in the CLIP score. This occurs because the operation of replacing partial latent disrupts the latent distribution and affects the representation of other parts of the image. Thanks to the rich semantic information in the latent, the drop is minor.

SRB	SIB	AdaIN	Sen. Acc	NED	CLIP Score
✗	✗	✗	29.12	56.99	0.3007
✓	✗	✗	36.22	61.17	0.3003
✓	✓	✗	36.93	64.35	0.3006
✓	✓	✓	**37.25**	**64.42**	**0.3027**

Table 5. **Ablation study.**

SIB without AdaIN. As illustrated in the second and third rows of Tab. 5, *Structure Injection Branch* further improves the accuracy. As similarly shown in the third column of Fig. 7, this branch can structurally improve the text structure, resulting in impressive improvement. However, when applied alone, it may disrupt image coherence, as shown in the second column and third row of Fig. 7, where the word 'Fun' occludes the eye. This issue arises from the lack of semantic information in the structural prior.

AdaIN Combination. As revealed by the third and fourth column of Fig. 7, AdaIN improves the accuracy and the harmony of the text and its background, lifting the CLIP score to a new level in the Tab. 5.

6. Conclusion

We advance visual text generation by tackling the challenge of complex text synthesis. Our proposed STGen introduces a dual-branch approach: the Semantic Rectification Branch, which refines text generation using latent extracted from simpler scenarios, and the Structure Injection Branch, which enhances text structure by incorporating latent of glyph image. For highly challenging cases, we break them into more manageable cases. Integrated via a dedicated ControlNet, STGen seamlessly enhances existing models. Extensive experiments on our benchmark confirm its superior performance, making STGen a promising step toward real-world applications.

References

[1] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. ediff-i: Text-to-image diffusion models with ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 1

[2] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *European conference on computer vision*, pages 707–723. Springer, 2022. 7

[3] Mehdi Boroumand, Mo Chen, and Jessica Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5): 1181–1193, 2019. 1

[4] Huiwen Chang, Han Zhang, Jarred Barber, Aaron Maschinot, José Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Patrick Murphy, William T. Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip Krishnan. Muse: Text-to-image generation via masked generative transformers. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, pages 4055–4075. PMLR, 2023. 2

[5] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser-2: Unleashing the power of language models for text rendering. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part V*, pages 386–402. Springer, 2024. 1, 3, 5, 6, 7

[6] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser: Diffusion models as text painters. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3, 5, 6, 7

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2

[8] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2315–2324, 2016. 3

[9] Adi Haviv, Shahar Sarfaty, Uri Hacohen, Niva Elkin-Koren, Roi Livni, and Amit H Bermano. Not every image is worth a thousand words: Quantifying originality in stable diffusion. *arXiv preprint arXiv:2408.08184*, 2024. 2

[10] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, 2021. 6

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 6840–6851, 2020. 2

[12] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017. 4

[13] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 5

[14] Praveen Krishnan, Rama Kovvuri, Guan Pang, Boris Vassilev, and Tal Hassner. Textstylebrush: transfer of text aesthetics from a single example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):9122–9134, 2023. 1

[15] Rosanne Liu, Dan Garrette, Chitwan Saharia, William Chan, Adam Roberts, Sharan Narang, Irina Blok, RJ Mical, Mohammad Norouzi, and Noah Constant. Character-aware models improve visual text rendering. *arXiv preprint arXiv:2212.10562*, 2022. 3

[16] Rosanne Liu, Dan Garrette, Chitwan Saharia, William Chan, Adam Roberts, Sharan Narang, Irina Blok, Rj Mical, Mohammad Norouzi, and Noah Constant. Character-aware models improve visual text rendering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16270–16297, 2023. 1, 3

[17] Zeyu Liu, Weicong Liang, Zhanhao Liang, Chong Luo, Ji Li, Gao Huang, and Yuhui Yuan. Glyph-byt5: A customized text encoder for accurate visual text rendering. *arXiv preprint arXiv:2403.09622*, 2024. 1, 3

[18] Zeyu Liu, Weicong Liang, Yiming Zhao, Bohan Chen, Ji Li, and Yuhui Yuan. Glyph-byt5-v2: A strong aesthetic baseline for accurate multilingual visual text rendering. *arXiv preprint arXiv:2406.10208*, 2024. 3

[19] Jian Ma, Mingjun Zhao, Chen Chen, Ruichen Wang, Di Niu, Haonan Lu, and Xiaodong Lin. Glyphdraw: Seamlessly rendering text with intricate spatial structures in text-to-image generation. *arXiv preprint arXiv:2303.17870*, 2023. 1, 3, 5

[20] Kento Masui, Mayu Otani, Masahiro Nomura, and Hideki Nakayama. Harnessing the latent diffusion model for training-free image style transfer. *arXiv preprint arXiv:2410.01366*, 2024. 4

[21] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021. 2

[22] Zhipeng Qian, Pei Zhang, Baosong Yang, Kai Fan, Yiwei Ma, Derek F Wong, Xiaoshuai Sun, and Rongrong Ji. Anytrans: Translate anytext in the image with large scale models. *arXiv preprint arXiv:2406.11432*, 2024. 3

[23] Yadong Qu, Qingfeng Tan, Hongtao Xie, Jianjun Xu, Yuxin Wang, and Yongdong Zhang. Exploring stroke-level modifications for scene text editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2119–2127, 2023. 1

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[25] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020. 3

[26] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021. 2

[27] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 5

[29] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. 1, 2, 3

[30] Dvir Samuel, Rami Ben-Ari, Nir Darshan, Haggai Maron, and Gal Chechik. Norm-guided latent space exploration for text-to-image generation. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[31] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 2

[32] Jeyasri Subramanian, Varnith Chordia, Eugene Bart, Shaobo Fang, Kelly Guan, Raja Bala, et al. Strive: Scene text replacement in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14549–14558, 2021. 1

[33] Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Geng, and Xuansong Xie. Anytext: Multilingual visual text generation and editing. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 1, 2, 3, 5, 6, 7, 8

[34] Qiangpeng Yang, Jun Huang, and Wei Lin. Swaptext: Image based texts transfer in scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14700–14709, 2020.

[35] Yukang Yang, Dongnan Gui, Yuhui Yuan, Weicong Liang, Haisong Ding, Han Hu, and Kai Chen. Glyphcontrol: Glyph conditional control for visual text generation. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3, 5, 6, 7

[36] Boqiang Zhang, Zuan Gao, Yadong Qu, and Hongtao Xie. How control information influences multilingual text image generation and editing? *arXiv preprint arXiv:2407.11502*, 2024. 3

[37] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 3

[38] Lingjun Zhang, Xinyuan Chen, Yaohui Wang, Yue Lu, and Yu Qiao. Brush your text: Synthesize any scene text on images via diffusion model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7215–7223, 2024. 1, 5, 6, 7

[39] Zhen Zhao, Jingqun Tang, Binghong Wu, Chunhui Lin, Shu Wei, Hao Liu, Xin Tan, Zhizhong Zhang, Can Huang, and Yuan Xie. Harmonizing visual text comprehension and generation. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. 3, 5, 6, 7

[40] Yuanzhi Zhu, Jiawei Liu, Feiyu Gao, Wenyu Liu, Xinggang Wang, Peng Wang, Fei Huang, Cong Yao, and Zhibo Yang. Visual text generation in the wild. *arXiv preprint arXiv:2407.14138*, 2024. 1, 3, 5, 6, 7