TextPecker: Rewarding Structural Anomaly Quantification for Enhancing Visual Text Rendering (Supplementary Materials)

A. Additional Results

We provide additional visualizations of manually annotated structural anomalies from diverse generative models and synthetic structural anomalies generated by our rendering engine in Fig. 8, evaluation samples of TextPecker in Fig. 9, and qualitative comparisons between Flux.1[dev] [13] and its RL-optimized variants in Fig. 1.

B. Additional Ablation Studies

Table 1. Ablation study on the effectiveness of reward design: PM for Pair-wise Matching, and SQ for Structural Quality reward, measured by TextPecker (InternVL3).

Generative Model	OCR Model	Settings			GenTextEval-EN	
		NED	PM	SQ	Qua.	Sem.
SD3.5-M [7]	-				0.671	0.265
	PP-OCRv5 [6]	✓			0.907	0.470
	PP-OCRv5 [6]	✓	✓		0.910	0.482
	TextPecker	✓	✓		0.956	0.498
	TextPecker	✓	✓	✓	**0.959**	**0.506**

We also provide additional ablation studies to deconstruct the reward function step-by-step and isolate the impact of each component, using StableDiffusion3.5-Medium [7] as the baseline, as shown in Tab. 1. Combining a conventional, structure-unaware OCR model with Pairwise Matching (PM) yields a 1.2% gain in semantic alignment, while structural quality sees a marginal 0.3% improvement—indicating PM enhances semantic feedback but minimally boosts structural fidelity without structural perception. Replacing the OCR model with TextPecker delivers gains across both dimensions (Sem. +1.6%, Qua. +4.6%), demonstrating the value of our structure-aware assessor. Finally, incorporating the structural quality term as an auxiliary reward brings further improvements (Sem. +0.8%, Qua. +0.3%) and achieves the best overall performance, confirming the synergy of the full TextPecker reward design.

C. Additional Generalization Results

We conduct **cross-model validation** on Gemini-2.5-flash-image [3] renderings to assess robustness under normal and extreme conditions (Tab. 2), and the results are consistent across these settings, with failures mainly on extremely stylized fonts where artistic deformations distort canonical glyph structure and blur the boundary between style and true structural errors (see Fig. 2, cases are all from Gemini). As for font variability, our dataset spans a large and diverse font pool across training and evaluation (Tab. 9)

Table 2. Robustness evaluation of TextPecker on Gemini-2.5-flash-image [3]: Performance under normal condition, extreme stylization, and low-contrast layouts.

Tab. A & Methods	Normal		Extreme Stylization		Low Contrast	
	TSAP-F1	CTR-R	TSAP-F1	CTR-R	TSAP-F1	CTR-R
InternVL-3-8B [28]	0.000	0.666	0.087	**0.588**	0.364	0.742
TextPecker-8B	**0.752**	**0.833**	**0.571**	0.577	**0.800**	**0.839**

Table 3. Additional Quantitative Comparisons of RL-Optimized Generative Models on **Chinese** Visual Text Benchxmarks (OneIG [1], LongText [9], GenTextEval) with multi reward setting. **O:** OCR Reward [15], **S:** TextPecker Semantic Reward, **Q:** TextPecker Structural Quality reward, **P:** Pickscore Reward [11], **A:** Aethetic Reward [17]. Results measurement and reward computation are both conducted by TextPecker (InternVL-3).

method	rewards	weights	OneIG		LongText		GenTextEval	
			Qua.	Sem.	Qua.	Sem.	Qua.	Sem.
Qwen-Image [23]	–	–	0.888	0.747	0.900	0.815	0.922	0.805
	OPA	7:1:2	0.898	0.788	0.912	0.845	0.944	0.859
	SQPA	5:2:1:2	**0.943**	**0.828**	**0.941**	**0.889**	**0.970**	**0.893**

D. Additional Results on RL for VTR

To further validate the efficacy of TextPecker and attain more robust performance in Visual Text Rendering, we conduct additional experiments under **a strengthened RL baseline setting**, with key design choices elaborated as follows:
Backbone enhancement. We adopt recent GRPO-related techniques[14, 21] to substantially enhance the efficiency and stability of the VTR optimization process, with implementation details supplemented in Sec. G:

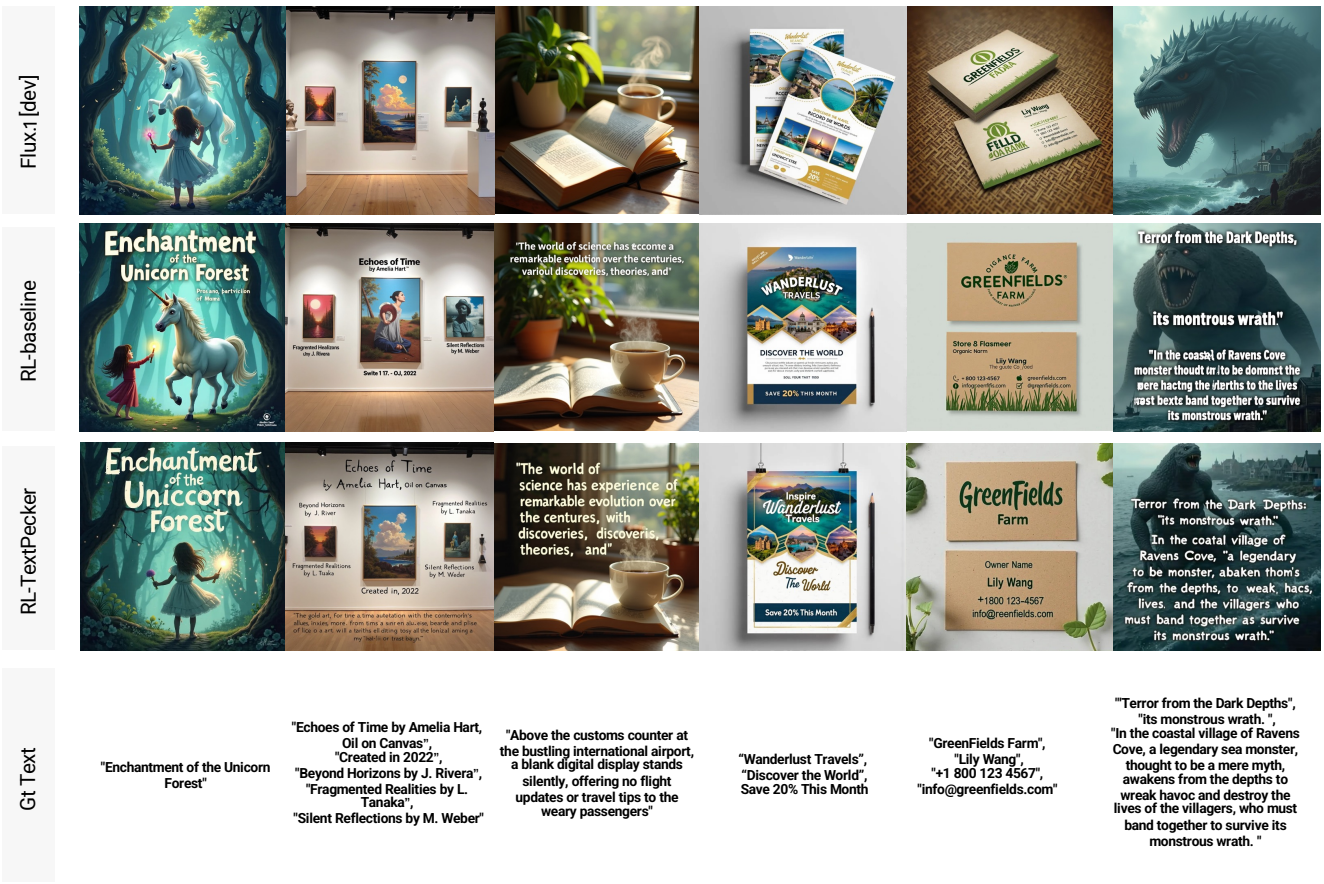


Figure 1. Qualitative Comparisons of Text Rendering for Flux.1[dev] [13] and RL-Optimized Variants.

(a) Extreme Stylization (b) Low Contrast

Figure 2. Hard Cases of TextPecker on Gemini-Rendered [3] Visual Text with Extreme Stylization and Low Contrast Layout.

(i) Flow-GRPO-Fast [14] is employed to accelerate training convergence by injecting stochasticity only on partial optimization steps instead of all steps;

(ii) GRPO-Guard [21] is employed to stabilize the training dynamics and mitigate implicit over-optimization issues in flow matching;

(iii) KL regularization enhancement (discussed in Flow-GRPO's [15] GitHub issues) is introduced to further alleviate over-optimization and reward hacking problems. The original formulation is:

$$D_{\mathrm{KL}}(\pi_\theta \parallel \pi_{\mathrm{ref}}) = \frac{\|\bar{x}_{t+\Delta t,\theta} - \bar{x}_{t+\Delta t,\mathrm{ref}}\|^2}{2\sigma_t^2 \Delta t}$$

To stabilize training dynamics and mitigate over-optimization more effectively, we redefine the KL divergence to operate over **velocity-based** policy distributions:

$$D_{\mathrm{KL}}(\pi_\theta \parallel \pi_{\mathrm{ref}}) = \|v_\theta(x_t, t) - v_{\mathrm{ref}}(x_t, t)\|^2$$

where all symbols follow the definitions in the Flow-GRPO [15] paper, with the core adjustment being the switch from state ($\bar{x}$) to velocity (v) as the regularization target.

Multi-Reward Regularization. In the main paper, we validated TextPecker's efficacy via experiments exclusive to text-rendering rewards. However, this single-reward setup inevitably degrades the model's aesthetic and image quality performance. To yield more robust VTR optimization, we propose a multi-reward regularization strategy: we augment the original TextPecker reward with PickScore [11] and Aesthetic Score [17], implicitly regularizing the VTR model to yield more robust RL optimization results.

We present quantitative results of TextPecker under this enhanced RL baseline in Tab. 3 and Tab. 4. Please note that all figures in the main paper and appendix are based on our original RL baseline, except for the additional visual comparisons between the two TextPecker variants provided in Fig. 3 and Fig. 4.

E. Details of Dataset

E.1. Details on Text-rich Image Generation

This section supplements the text-rich image generation dataset construction details in the main text. We present detailed statistics on the number of generated images categorized by language and various generative models employed. English dataset statistics in Tab. 5 and Chinese dataset statistics shown in Tab. 6.

E.2. Details on Synthetic Data Augmentation

As demonstrated in the main paper, models trained solely on manually annotated data generalize poorly to unseen structural anomalies. This limitation is particularly acute for Chinese characters, whose 2D structure and vast inventory (8,000 common characters) create a combinatorial explosion of anomalies impossible to annotate exhaustively. To overcome this, we extend the SynthTIGER [25] renderer with two key enhancements: (1) image-level layout arrangements to simulate complex scenes, (2) Structural Anomaly Construction engine tailored to systematically generate diverse structural errors in Chinese.

Key parameters for our rendering engine, covering both canonical text generation and structural anomaly construction, are detailed in Table 9. Our parameter choices are guided by the goal of training precise structural perception, not robust text extraction. Consequently, to preserve clear structural features, we intentionally disabled heavy post-processing (e.g., noise, blur) and certain style effects (e.g., extrusion), while limiting geometric transformations (e.g., skew, rotation) to moderate ranges. Notably, we have **a font pool of 976 types** to enhance font diversity. This ensures the rendered text maintains high structural clarity amidst realistic diversity, which is critical for our training objective.

E.3. Structural Anomaly Perception Test Set

We provide detailed statistics of the structural perception test set in Tab. 7. To further validate the fairness and effectiveness of our results, we additionally conduct evaluations on a **real-only** test split (all synthetic samples excluded), with results presented in Tab. 8.

E.4. Statistics of the RL Prompt Set for VTR

We present the statistics of the curated prompt set used for RL-based VTR optimization in Fig. 5. The prompt set is designed to encompass diverse text lengths and content for effective reinforcement learning.

For English text rendering, we curate prompts from TextAtlas5M [20], ensuring a rich and varied dataset. For Chinese text rendering, we adopt a similar paradigm as described in the main paper, starting with a comprehensive text corpus sampled from WanJuan1.0 [10], which covers a wide range of modern Chinese common characters. Additionally, we use Qwen3-235B-A22B [24] to generate diverse style descriptions of fonts. These style descriptions are integrated with the corpus to create the final prompt set. The statistics are visualized in Fig. 5.

E.5. Statistics of the GenTextEval Dataset

To facilitate re-evaluation with TextPecker and build upon the strengths of existing benchmarks, we construct a dataset named GenTextEval, which integrates English and Chinese prompts from multiple sources [1, 5, 9, 20]. In light of the limited availability of Chinese-rendering benchmarks [1, 9, 23] (with ChineseWord remaining unavailable for open-source at the time of our experiments), this dataset is further enriched with Chinese prompts curated as described in Sec. E.4. The final GenTextEval dataset comprises 314 prompts for English rendering and 417 prompts for Chinese rendering. Following the traditional paradigm, each prompt generates four distinct image outputs to ensure fairer assessment. We offer a statistical overview of the GenTextEval dataset in Fig. 6.

F. Prompt Template for TextPecker

We present the prompting templates used for TextPecker's training and testing in Fig. 7. To ensure consistency and comparability across evaluations, we adopt the identical template for all other MLLM baselines.

G. Additional Implementation Details

This section provides further implementation details, supplementing the overview in the main paper. We employ the Flow-GRPO [15] framework for all Reinforcement Learning (RL) based VTR optimization experiments. Notably, Flow-GRPO is an actively evolving repository, and the implementations reported here reflect the stable version available at the time of our experiments. As noted in the main paper, the overall methodology adheres to Flow-GRPO's core design, while the specific hyperparameters are carefully tuned for each base model (building upon the framework's default configurations) to ensure stable and effective training. The resolution for all generated images is set to 512×512 pixels. The model-specific configurations are detailed below.

SD3.5-M [7]: We use 30 sampling steps for training and 40 for evaluation. The noise level is set to 0.8, and the guidance scale is 1.0 (following the Flow-GRPO-Fast framework,

Table 4. Additional Quantitative Comparisons of RL-Optimized Generative Models on **English** VTR Benchmarks (OneIG [1], LongText [9], CVTG [5], GenTextEval, TIIF [22], TextAtlas [20], LeX [26]) with multi reward setting. **O:** OCR Reward [15], **S:** TextPecker Semantic Reward, **Q:** TextPecker Structural Quality reward, **P:** Pickscore Reward [11], **A:** Aethetic Reward [17]. Results measurement and reward computation are both conducted by TextPecker (InternVL-3).

method	rewards	weights	OneIG		LongText		CVTG		GenTextEval		TIIF		TextAtlas		LeX	
			Qua.	Sem.	Qua.	Sem.	Qua.	Sem.	Qua.	Sem.	Qua.	Sem.	Qua.	Sem.	Qua.	Sem.
SD3.5-M [7]	–	–	0.840	0.507	0.836	0.407	0.843	0.466	0.666	0.262	0.758	0.347	0.646	0.269	0.810	0.454
	OPA	7:1:2	0.908	0.588	0.913	0.508	0.895	**0.621**	0.896	0.461	0.886	0.483	0.916	0.436	0.894	0.563
	SQPA	5:2:1:2	**0.940**	**0.607**	**0.959**	**0.534**	**0.926**	0.587	**0.954**	**0.519**	**0.941**	**0.506**	**0.954**	**0.462**	**0.940**	**0.591**
Flux.1[dev] [13]	–	–	0.870	0.578	0.925	0.584	0.889	0.510	0.664	0.332	0.933	0.540	0.683	0.307	0.946	0.667
	OPA	7:1:2	0.977	0.739	0.977	0.763	0.974	0.780	0.982	0.739	0.986	0.719	0.983	0.640	0.988	0.741
	SQPA	5:2:1:2	**0.990**	**0.775**	**0.992**	**0.780**	**0.993**	**0.824**	**0.991**	**0.762**	**0.991**	**0.735**	**0.993**	**0.649**	**0.991**	**0.807**
Qwen-Image [23]	–	–	0.954	0.812	0.961	0.831	0.960	0.817	0.958	0.723	0.933	0.682	0.953	0.665	0.927	0.760
	OPA	7:1:2	0.963	0.840	0.967	0.858	0.962	0.848	0.974	0.808	0.964	0.764	0.970	0.728	0.958	0.850
	SQPA	5:2:1:2	**0.983**	**0.888**	**0.982**	**0.891**	**0.976**	**0.889**	**0.990**	**0.876**	**0.975**	**0.800**	**0.982**	**0.746**	**0.968**	**0.883**

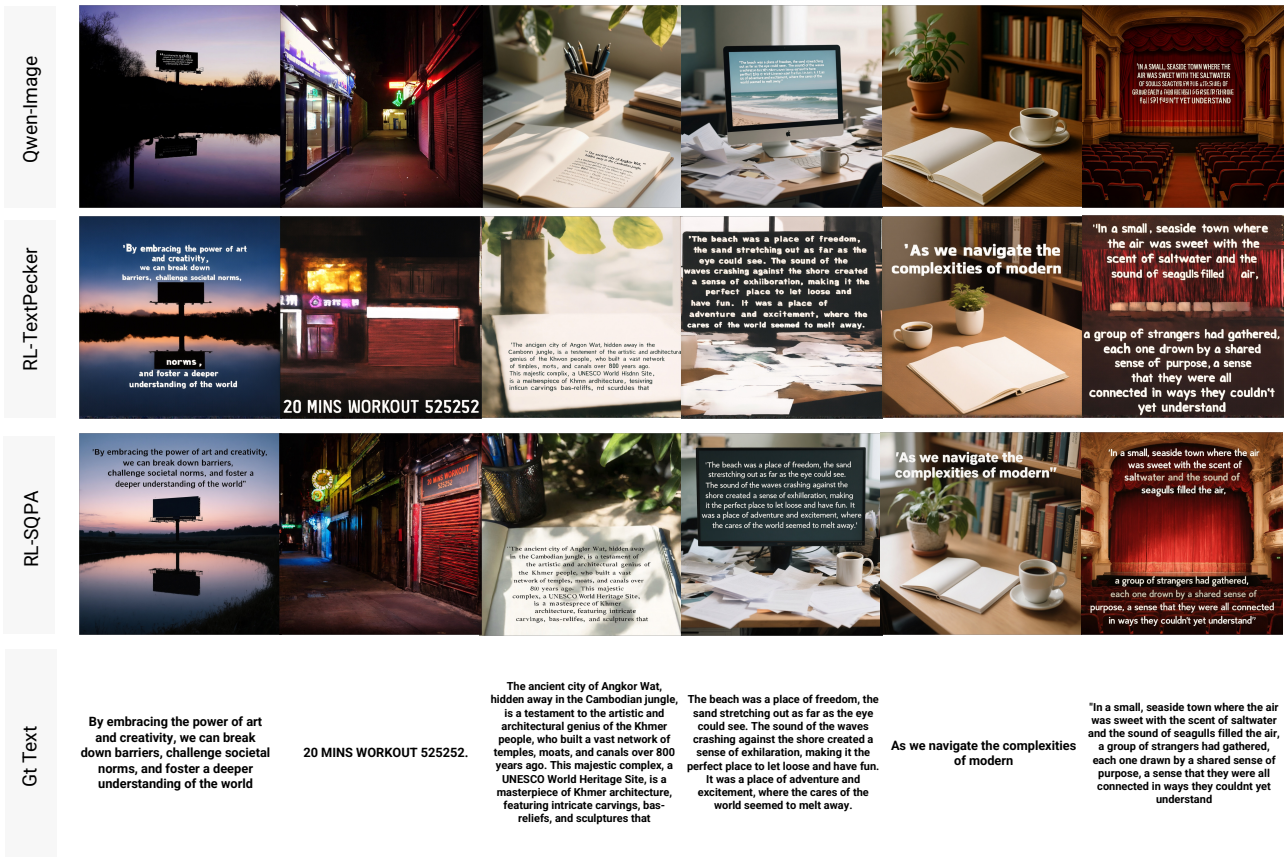


Figure 3. Qualitative comparisons of text rendering results (**English**) among different RL baseline settings. RL-TextPecker denotes the RL setting in the main paper, and RL-SQPA refers to our enhanced RL setting as described in Sec. D.

CPS sampling and No-CFG are adopted to improve training efficiency). The KL ratio β is 0.04. For LoRA, we adopt a rank r of 32 and an alpha α of 64. Training is conducted using the Flow-GRPO-Fast framework.

Flux.1[dev] [13]: We set 14 sampling steps for training and 28 for evaluation. The noise level is 0.9, the guidance scale is 3.5, and the KL ratio β remains 0.04. For LoRA, we use a rank r of 64 and an alpha α of 128. No Flow-GRPO variant

Figure 4. Qualitative comparisons of text rendering results (**Chinese**) among different RL baseline settings. RL-TextPecker denotes the RL setting in the main paper, and RL-SQPA refers to our enhanced RL setting as described in Sec. D.

Table 5. Statistics of English text-rich images generated by different models, labeled at box and image levels. Proportions are computed over all instances.

Model	Level	Samples	Proportion
AnyText [19]	Box-level	30647	7.38%
	Image-level	7405	1.78%
Flux.1[dev] [13]	Box-level	91705	22.07%
	Image-level	19181	4.62%
Qwen-Image [23]	Box-level	20308	4.89%
	Image-level	2647	0.64%
SD3 [7]	Box-level	105725	25.45%
	Image-level	17766	4.28%
SDv15 [16]	Box-level	61961	14.91%
	Image-level	6033	1.45%
SeedDream3.0 [8]	Box-level	47921	11.53%
	Image-level	4144	1.00%

is employed for this model.

Qwen-Image [23]: We employ 10 sampling steps for training and 50 for evaluation. The noise level is set to 1.2, with a guidance scale of 4.0 and a KL ratio β of 0.004. For LoRA, we adopt a rank r of 64 and an alpha α of 128. Training is conducted using the Flow-GRPO-Fast framework.

Table 6. Statistics of Chinese text-rich images generated by different models, labeled at box and image levels. Proportions are computed over all instances.

Model	Level	Samples	Proportion
CogView4 [4]	Box-level	36000	11.14%
	Image-level	17005	5.26%
Kolors [12]	Box-level	35549	10.99%
	Image-level	19312	5.98%
Qwen-Image [23]	Box-level	26597	8.23%
	Image-level	6225	1.93%
SeedDream3.0 [8]	Box-level	146395	45.31%
	Image-level	36032	11.15%

G.1. Computational cost and latency.

The evaluator is used only during RL training and run as a separate asynchronous service, hence it adds negligible overhead and does not affect inference latency; on SD3.5-

Table 7. Statistics of our constructed text-rich image structural perception test dataset with structural-anomaly labels at box and image levels. Proportions are computed over all instances.

Data Type	Level	Samples	Proportion
Manual Annotations	Box	444	41.85%
	Image	417	39.29%
Synthetic Anomaly Text	Box	50	4.71%
	Image	50	4.71%
Synthetic Normal Text	Box	50	4.71%
	Image	50	4.71%
Total	–	1061	100%

Table 8. Performance of TextPecker on Real-only Test Splits

Methods	Chinese				English			
	Image		Box		Image		Box	
	TSAP-F1	CTR-R	TSAP-F1	CTR-R	TSAP-F1	CTR-R	TSAP-F1	CTR-R
InternVL3-8B (Baseline)	0.106	**0.955**	0.244	0.791	0.183	0.759	0.304	0.570
TextPecker-8B (Anno)	0.866	0.849	0.906	0.815	**0.874**	**0.938**	0.809	0.918
TextPecker-8B (Anno + Syn)	**0.901**	0.917	**0.955**	**0.995**	0.850	0.931	**0.840**	**0.944**

M[7], 100 RL steps take 5.52 h (TextPecker) vs. 5.40 h (PPOCRv5[6]).

H. Additional Implementation Details on Sec. D

As mentioned in Sec. D, we conducted additional experiments utilizing an enhanced RL baseline. This baseline incorporates several advanced techniques including Flow-GRPO-Fast [14], GRPO-Guard [21], Velocity KL loss, and multi-reward regularization. This section provides the specific hyperparameter details for experiments in Tab. 3 and Tab. 4. LoRA configurations remain identical to those described in Sec. D.

SD3.5-M [7]: We use 40 sampling steps for both training and evaluation. An SDE window of size 12 is applied during the first half of the sampling process. The key hyperparameters are set as follows: a noise level of 0.9, a guidance scale of 4.5, and a learning rate of 10^{-4}. The ratio β for the Velocity KL loss is set to 10^{-4}, and the clipping range is set to 2×10^{-6}.

Flux.1[dev] [13]: We set the number of sampling steps to 28 for both training and evaluation. An SDE window of size 9 is used in the first half of the sampling steps. The noise level is 0.9, the guidance scale is 3.5, and the learning rate is 10^{-4}. The Velocity KL loss ratio β is configured to 10^{-4}, and the clipping range is set to 2×10^{-6}.

Qwen-Image [23]: We employ 20 sampling steps for training and 50 for evaluation. During training, an SDE window of size 5 is applied to the initial half of the sampling steps. The noise level is set to 1.2, the guidance scale is 4, and the learning rate is 10^{-4}. The Velocity KL ratio β is 10^{-3}, and the clipping range is set to 2×10^{-5}.

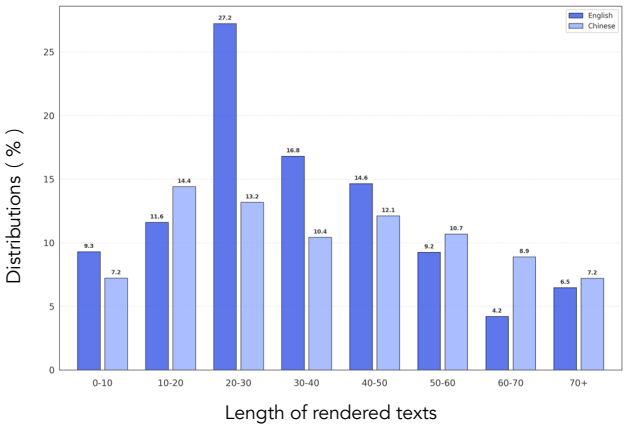


Figure 5. Statistics of the RL prompt set for RL-based VTR optimization (English: word-level; Chinese: character-level).

I. Limitations

TextPecker paves a novel path for addressing the core bottleneck in VTR evaluation and RL-based optimization, leveraging a structural-anomaly-aware RL reward that delivers complementary signals for semantic alignment and structural quality. While providing a foundational step towards structurally faithful VTR, our work still has several limitations that point to meaningful directions to be explored.

First, our structural anomaly synthesis is contingent upon the availability of stroke-level font data. This dependency currently restricts its application to standard fonts, precluding the generation of anomalies in artistic or proprietary typefaces lacking such data.

Second, our work is currently confined to Chinese and English text rendering, with efficient multilingual extension as a key area for future exploration.

Third, our TextPecker evaluator is equipped with box-level perception ability, which theoretically enables it to support downstream VTR-related tasks such as text translation and local text editing, which are often challenging for general editing methods [2, 13, 18, 19, 27]. Validating the effectiveness of evaluation and RL optimization on these downstream tasks is left for future work.

Fourth, challenges arise in handling artistic text generation (see Fig. 2 above), which is an increasingly demanded scenario. Artistic text often involves deliberate modifications to standard structures, such as connected strokes, added symbols, or pictorial variations, making it inherently difficult to define a single standard or ground truth. Furthermore, artistic designs are continuously evolving, presenting a moving target that conflicts with the structural consistency objectives of our current framework. Addressing the evaluation and optimization of artistic text generation remains a challenging yet impactful research direction, necessitating the integration of creative expression with principles of structure-aware

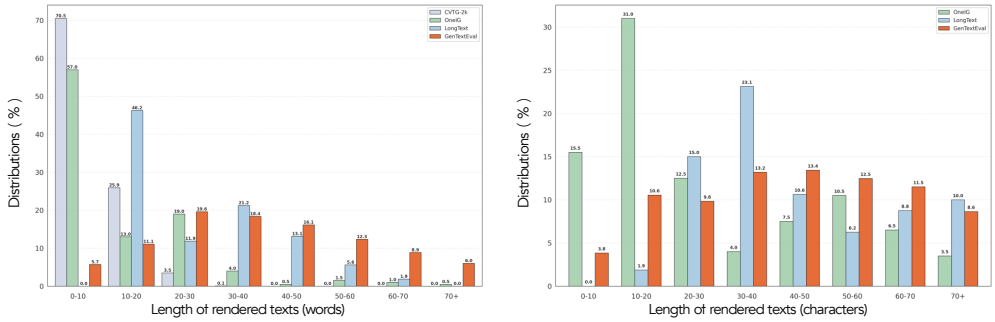


Figure 6. Comparison among CVTG-2K[5], OneIG-Bench[1], LongText-Bench[9], and Our Proposed GenTextEval-Bench with Respect to the Length of Rendered Texts in English (Left) and Chinese (Right).

Image-level Prompt for TextPecker

Query: This is a text-generated image. Please recognize all visible text in the entire image. Marking rules: 1. Use <#> for structurally flawed (e.g., extra/missing strokes, distortion) unrecognizable Chinese characters or single English letters; 2. Use <###> exclusively for structurally flawed unrecognizable single English words (not multi-word phrases, lines, or sentences). Output in the following JSON format: {"recognized_text": "All text in the image (including structural error markers)"}

Box-level Prompt for TextPecker

Query: This is a text-generated image. Please recognize all visible text in the local area "bbox_2d:[x1, y1, x2, y2]". Marking rules: 1. Use <#> for structurally flawed (e.g., extra/missing strokes, distortion) unrecognizable Chinese characters or single English letters; 2. Use <###> exclusively for structurally flawed unrecognizable single English words (not multi-word phrases, lines, or sentences). Output in the following JSON format: {"recognized_text": " Text in "bbox_2d:[x1, y1, x2, y2]" (including structural error markers)"}

Figure 7. The prompting template used for our TextPecker.

textual modeling.

Table 9. Key parameters for canonical text rendering and structural anomaly construction.

Parameter Category	Canonical Chinese Text	Canonical English Text	Structural Anomaly Construction
Basic Text Configuration			
Vertical text probability	10%	10%	10%
Number of elements per sample	3–10	3–10	3–10
Text length range	1–25 characters	3–25 characters	1–25 characters
Font Settings			
Number of font types	976	976	976
Font size range	50–100 pt	50–100 pt	50–100 pt
Layout Parameters			
Horizontal spacing between elements	50–200 px	50–200 px	50–200 px
Vertical line spacing	10–20 px	10–20 px	10–20 px
Length ratio range	0.8–1.0	0.8–1.0	0.8–1.0
Random offset probability	20%	20%	20%
Random offset range	10–30 px	10–30 px	10–30 px
Image margin	15 px	15 px	15 px
Flow layout probability	80%	80%	80%
Curve layout probability	20%	20%	20%
Style Effects			
Style application probability	25%	25%	25%
- Text border (probability)	100%	100%	100%
Size ratio	5–15%	5–15%	5–15%
Alpha	1.0	1.0	1.0
- Text shadow (probability)	0%	0%	0%
- Text extrusion (probability)	0%	0%	0%
Geometric Transformation			
Transformation application probability	50%	50%	50%
- Perspective x (weight)	1	1	1
Percents	0.8	0.8	0.8
- Perspective y (weight)	1	1	1
Percents	0.8–1	0.8–1	0.8–1
- Trapezoidate x (weight)	1	1	1
Percent	0.8–1	0.8–1	0.8–1
- Trapezoidate y (weight)	1	1	1
Percent	0.8–1	0.8–1	0.8–1
- Skew x (weight)	2	2	2
Angle	0–30°	0–30°	0–30°
- Skew y (weight)	2	2	2
Angle	0–10°	0–10°	0–10°
- Rotate (weight)	3	3	3
Angle	0–10°	0–10°	0–10°
Structural Anomaly Generation			
Anomaly generation probability	0%	0%	50%
- Deletion (probability)	–	–	40%
- Insertion (probability)	–	–	40%
- Swapping (probability)	–	–	40%

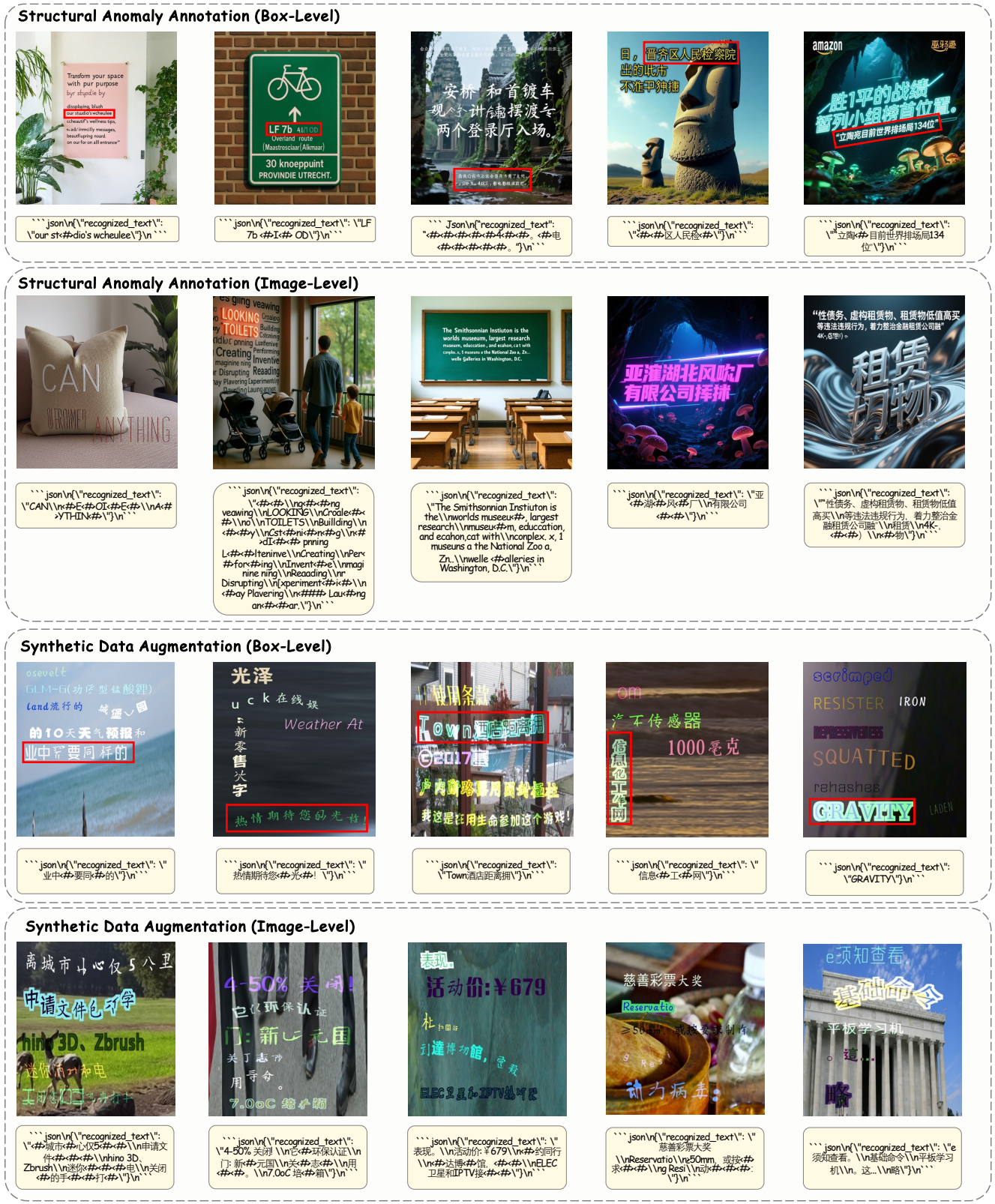
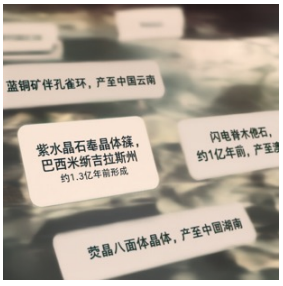


Figure 8. This figure shows manually annotated structural anomalies (box-level and image-level) from various generative models, alongside synthetic structural anomalies generated by our rendering engine for data augmentation.

"prompt": "一幅充满奇思妙想的漫画风格插图，描绘两位办公室员工在公园长椅上共进午餐时的幽默互动。对话框自然地按照对话流程排列。画面左上角，第一位员工的对话框内容为：“我带了一份新鲜沙拉，开始健康生活！”右上角第二位员工的对话框位置，其略显怀疑的表情下写着：“早上会议时你不是在吃甜甜圈吗？”左下角第一位员工防御性地回应：“平衡是关键！绿叶蔬菜可以抵消甜甜圈的糖分。”右下角第二位员工带着笑意和调侃的语气说：“啊，我搞错了。你应该给HR发邮件，建议推出新的甜甜圈沙拉健康计划。”诙谐的面部表情伴随着机智的对话，对话框以对角线排列方式自然引导观者体验这场幽默互动。",
"target": "我带了一份新鲜沙拉，开始健康生活！ 早上会议时你不是在吃甜甜圈吗？ 平衡是关键！ 绿叶蔬菜可以抵消甜甜圈的糖分。 啊，我搞错了。 你应该给HR发邮件，建议推出新的甜甜圈沙拉健康计划。",
"pecker_qua": 0.8888888888888888,
"pecker_sem": 0.5802469135802469,
"recognized_text": "早上会议时你不是在 我带了一份新鲜沙 吃甜甜圈吗 开始健康生活 啊 我搞错了 你<#><#>HR 平衡是关键 绿叶蔬菜 可以抵涂甜甜圈的糖分 <#><#><#><#><#><#><#><#>"

"prompt": "阳光明媚的一天，繁忙的游乐园入口处挂满了五彩缤纷的横幅。入口上方悬挂着一条醒目的横幅，用清晰粗体的字母写着：“欢迎来到冒险王国，乐趣永无止境！”在问候语下方，稍小一些的文字写着：“体验惊险刺激的游乐设施、美味可口的美食和难忘的家庭时光”。售票处附近的小型横幅用易读的文字标明：“工作日折扣：12岁以下儿童五折！”以及“年卡持有者专属通道”。其他附近标识牌显示着“园区每日上午10点至晚上9点开放”等信息。游客们稍作停留，阅读这些引人注目且充满吸引力的文字信息后，兴奋地朝公园入口走去。",
"target": "欢迎来到冒险王国，乐趣永无止境！ 体验惊险刺激的游乐设施、美味可口的美食和难忘的家庭时光 工作日折扣：12岁以下儿童五折！ 年卡持有者专属通道 园区每日上午10点至晚上9点开放",
"pecker_qua": 0.9864864864864865,
"pecker_sem": 0.8227848101265822,
"recognized_text": "欢迎来到冒险王国 乐趣永无止境 体验惊险刺激的游乐设施、 美味可口的美食和难忘的家庭时光 工作日折扣： 12岁以<#>儿童五折 <#>卡每日上午10点至 <#>晚上9属通道"

"prompt": "A humorous cartoon-style comic panel set inside an art museum, depicting a confused visitor engaging in playful dialogue with the enthusiastic tour guide about an abstract painting. Speech bubbles are positioned naturally to guide viewers easily through their conversation. At the upper-left, the puzzled visitor's speech bubble reads, \"Are you sure this upside-down triangle represents humanity's struggle?\" In the top-right corner, the smiling guide replies confidently, \"Precisely! Humanity often struggles between deep meaning and pizza cravings\". Lower-left corner speech bubble shows the visitor responding thoughtfully, \"Well, in that case, I deeply resonate with this masterpiece\". Finally, bottom-right side next to the guide, a cheerful bubble summarizes, \"Excellent! True art speaks clearly to those who hunger for understanding\". The humorous expressions and clear diagonal speech bubble arrangement naturally guide viewers through the amusing exchange from top-left to bottom-right.",
"target": "Are you sure this upside-down triangle represents humanitys struggle? Precisely! Humanity often struggles between deep meaning and pizza cravings Well, in that case, I deeply resonate with this masterpiece Excellent! True art speaks clearly to those who hunger for understanding",
"pecker_qua": 0.9915966386554622,
"pecker_sem": 0.4777591036414566,
"recognized_text": "THE YOU OFF ARE WrAMGuS STRuGGLE, SUEE THIS THEYAI IE PESINOUT UPSIDE-DOWN HUMANITY,G<#><#>ides. STRVGGLES DEEP MEANING AND PIZZA CRAVINGS? WELL, IN THAT CASE, I DEEPLY REMOTANTE WITH THIS MASTERPIECES EXPELLENT! \"TRUE ART CLEARLY\" NEXTES TO THOSE WHO HUNGER FOR UNDERSTANDING!"



"prompt": "一个精心陈列的矿物和晶体标本的玻璃展柜。每个标本都配有详细且易于阅读的信息标签；其中一个显眼的标签清晰标注着“紫水晶石英晶体簇，巴西米纳斯吉拉斯州”，下方附有更小字号的说明文字明确写着“约1.3亿年前形成”。附近的其他信息标签精准标注着诸如“蓝铜矿伴孔雀石，产自中国云南”、“闪电脊木化石，约1亿年前，产自澳洲”以及“萤石八面体晶体，产自中国湖南”等展品信息。这些印刷整齐、简洁明了的标签有效吸引并教育着观众，增强他们在欣赏展示矿物标本时的互动体验与理解深度。",
"target": "紫水晶石英晶体簇，巴西米纳斯吉拉斯州 约1.3亿年前形成 蓝铜矿伴孔雀石，产自中国云南 闪电脊木化石，约1亿年前，产自澳洲 萤石八面体晶体，产自中国湖南",
"pecker_qua": 0.9333333333333333,
"pecker_sem": 0.44285714285714284,
"recognized_text": "蓝铜矿伴孔雀环 产至中国云南 闪电脊木<#>石 紫水晶石奉晶体簇 约1亿年<#> 产至<#> 巴西米<#>吉拉斯州 约1.3亿年前形成 荧晶八面体晶体 产至中<#>湖南"

"prompt": "A lively, crowded outdoor marketplace fills a sunny plaza, bustling energetically with shoppers weaving among vibrant vendor stands overflowing with fresh, colorful goods. Drawing clear attention at the heart of this dynamic market scene is a large, rustic wooden sign prominently hung above a popular stall, displaying attractive, easily readable lettering that warmly announces \"Farm Fresh & Locally Produce\". Beneath this main headline, elegant yet simple text proclaims encouraging messages, clearly stating \"Taste Nature's Best Support Local Farmers!\" and the enticing incentive \"Special Offer: Organic 10% Off Today Only!\" Scattered artfully around the stall are smaller, charmingly handwritten chalkboard-style signs clearly displaying inviting additional notes such as \"Fresh Apples, Strawberries and Seasonal Veggies Available\" further engaging and drawing in curious visitors. Shoppers frequently pause to thoughtfully read and appreciate these inviting, vividly presented signs, adding warmth and authenticity to the overall bustling marketplace atmosphere.",
"target": "Farm Fresh & Locally Produce Taste Natures Best Support Local Farmers! Special Offer: Organic 10% Off Today Only! Fresh Apples, Strawberries and Seasonal Veggies Available",
"pecker_qua": 1.0,
"pecker_sem": 0.71,
"recognized_text": "Farm Fresh & Locally Produce. Taste Nature's Best Support Local Farmers! Special Offer: Organic 10% Off Today Only!"

Figure 9. This figure presents evaluation samples of TextPecker, showcasing its performance in detecting structural anomalies across diverse text rendering scenarios. $\omega = 1$ for structural quality score visualization.

References

[1] Jingjing Chang, Yixiao Fang, Peng Xing, Shuhan Wu, Wei Cheng, Rui Wang, Xianfang Zeng, Gang Yu, and Hai-Bao Chen. Oneig-bench: Omni-dimensional nuanced evaluation for image generation. *arXiv preprint arxiv:2506.07977*, 2025. 1, 3, 4, 7

[2] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 6

[3] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 1, 2

[4] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *NIPS*, 34:19822–19835, 2021. 5

[5] Nikai Du, Zhennan Chen, Shan Gao, Zhizhou Chen, Xi Chen, Zhengkai Jiang, Jian Yang, and Ying Tai. Textcrafter: Accurately rendering multiple texts in complex visual scenes. *arXiv preprint arXiv:2503.23461*, 2025. 3, 4, 7

[6] Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, et al. Pp-ocr: A practical ultra lightweight ocr system. *arXiv preprint arXiv:2009.09941*, 2020. 1, 6

[7] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 1, 3, 4, 5, 6

[8] Yu Gao, Lixue Gong, Qiushan Guo, Xiaoxia Hou, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, et al. Seedream 3.0 technical report. *arXiv preprint arXiv:2504.11346*, 2025. 5

[9] Zigang Geng, Yibing Wang, Yeyao Ma, Chen Li, Yongming Rao, Shuyang Gu, Zhao Zhong, Qinglin Lu, Han Hu, Xiaosong Zhang, et al. X-omni: Reinforcement learning makes discrete autoregressive image generative models great again. *arXiv preprint arXiv:2507.22058*, 2025. 1, 3, 4, 7

[10] Conghui He, Zhenjiang Jin, Chao Xu, Jiantao Qiu, Bin Wang, Wei Li, Hang Yan, Jiaqi Wang, and Dahua Lin. Wanjuan: A comprehensive multimodal dataset for advancing english and chinese large models. *arXiv preprint arXiv:2308.10755*, 2023. 3

[11] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023. 1, 2, 4

[12] Kwai-Kolors Kolors. Kolors. https://huggingface.co/Kwai-Kolors/Kolors, 2025. Accessed: 2025-09-22. 5

[13] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 1, 2, 4, 5, 6

[14] Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Miles Yang, and Zhao Zhong. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025. 1, 2, 6

[15] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025. 1, 2, 3, 4

[16] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10674–10685, 2022. 5

[17] Christoph Schuhmann. Laion-aesthetics. *LAION Blog*, 2022. 1, 2, 4

[18] Yichun Shi, Peng Wang, and Weilin Huang. Seededit: Align image re-generation to image editing. *arXiv preprint arXiv:2411.06686*, 2024. 6

[19] Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Geng, and Xuansong Xie. Anytext: Multilingual visual text generation and editing. *arXiv preprint arXiv:2311.03054*, 2023. 5, 6

[20] Alex Jinpeng Wang, Dongxing Mao, Jiawei Zhang, Weiming Han, Zhuobai Dong, Linjie Li, Yiqi Lin, Zhengyuan Yang, Libo Qin, Fuwei Zhang, Lijuan Wang, and Min Li. Textatlas5m: A large-scale dataset for dense text image generation. *arXiv preprint arXiv:2502.07870*, 2025. 3, 4

[21] Jing Wang, Jiajun Liang, Jie Liu, Henglin Liu, Gongye Liu, Jun Zheng, Wanyuan Pang, Ao Ma, Zhenyu Xie, Xintao Wang, et al. Grpo-guard: Mitigating implicit over-optimization in flow matching via regulated clipping. *arXiv preprint arXiv:2510.22319*, 2025. 1, 2, 6

[22] Xinyu Wei, Jinrui Zhang, Zeqing Wang, Hongyang Wei, Zhen Guo, and Lei Zhang. Tiif-bench: How does your t2i model follow your instructions? *arXiv preprint arXiv:2506.02161*, 2025. 4

[23] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. 1, 3, 4, 5, 6

[24] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 3

[25] Moonbin Yim, Yoonsik Kim, Han-Cheol Cho, and Sungrae Park. Synthtiger: Synthetic text image generator towards better text recognition models. In *International conference on document analysis and recognition*, pages 109–124. Springer, 2021. 3

[26] Shitian Zhao, Qilong Wu, Xinyue Li, Bo Zhang, Ming Li, Qi Qin, Dongyang Liu, Kai Zhang, Hongsheng Li, Yu Qiao, Peng Gao, Bin Fu, and Zhen Li. Lex-art: Rethinking text generation via scalable high-quality data synthesis. *arXiv preprint arXiv:2503.21749*, 2025. 4

[27] Hanshen Zhu, Zhen Zhu, Kaile Zhang, Yiming Gong, Yuliang Liu, and Xiang Bai. Training-free geometric image editing on

diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19130–19140, 2025. 6

[28] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. 1