LENS: RETHINKING TRAINING EFFICIENCY FOR FOUNDATIONAL TEXT-TO-IMAGE MODELS

Anonymous authors
Paper under double-blind review



Figure 1: Images generated by *Lens* at 1440 resolution. Appendix C.1 provides more visualizations.

ABSTRACT

Training foundational text-to-image (T2I) models typically requires massive computational resources. We introduce *Lens*, a 3.8B-parameter T2I model that achieves performance competitive with, and in several cases surpassing, state-of-the-art models with more than 6B parameters across various benchmarks, while requiring significantly less training compute. For example, *Lens* requires only about 19.3% of the training compute used by Z-Image. The training efficiency of *Lens* stems from two key strategies beyond its compact model size. First, we maximize *data information density per training batch* by (i) training on *Lens-800M*, a dataset of 800M densely captioned image-text pairs whose captions are generated by GPT-4.1 and contain approximately 109 words on average, providing richer semantic supervision than conventional short captions, and (ii) constructing each batch from images with multiple resolutions and diverse aspect ratios, thereby enlarging the effective visual coverage of each optimization step. Second, we improve *convergence speed* through careful architectural choices, including adopting a semantic VAE that provides better latent representations and employing a strong language encoder that accelerates optimization while enabling multilingual

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

generalization from English-only training data. After pre-training, we apply reinforcement learning with taxonomy-driven prompts (*Lens-RL-8K*) and structured reward rubrics to suppress artifacts and improve visual quality, a reasoner module with training-free system prompt search to better align user requests with the model, and distillation-based acceleration for 4-step inference. Through efficient training and systematic optimization, *Lens* generalizes to arbitrary aspect ratios from 1:2 to 2:1 and resolutions up to 1440^2, and supports prompts in several commonly used languages. Thanks to its compact size, *Lens* generates a 1024^2 image in 3.15 seconds on a single NVIDIA H100 GPU, while its distilled turbo version performs 4-step generation in 0.84 seconds. We will release both the pre-trained and post-trained models.

1 INTRODUCTION

Recent advances in foundational text-to-image (T2I) generative models have demonstrated remarkable capabilities in high-fidelity image synthesis and complex prompt understanding, as discussed in Appendix A. However, these gains have come at a substantial cost: training such models typically requires massive computational resources, leading to prohibitive financial and environmental expenses. For example, Z-Image (Cai et al., 2025a) requires approximately 314K H800 GPU hours for pre-training, highlighting the growing scalability challenge of training foundation-scale T2I models.

In this paper, we focus on improving the **training-time efficiency** of foundational T2I models. We argue that training-time efficiency is jointly determined by three key factors: (1) *model size*, which directly affects the computational cost of each training step; (2) *data information density per training batch*, which determines how much useful supervision the model can extract from each update; and (3) *convergence speed*, which determines the overall number of training iterations, as faster convergence enables the model to achieve strong performance with fewer optimization steps. Therefore, improving training-time efficiency requires not only reducing model scale, but also increasing the learning value of each batch and accelerating convergence throughout training.

Motivated by these factors, we introduce **Lens**, a foundational T2I model designed for efficient training. First, to reduce the per-step computational cost, we constrain *Lens* to 3.8B parameters. In contrast, recent state-of-the-art open-source models, including Z-Image (6B) (Cai et al., 2025a), LongCat-Image (6B) (Team et al., 2025), FLUX.2 (9B) (Labs, 2025), Qwen-Image (20B) (Wu et al., 2025), and Hunyuan-Image-3.0 (MoE, 80B) (Cao et al., 2025), operate at scales of 6B parameters or larger. Despite its relatively compact 3.8B-parameter scale, *Lens* achieves performance competitive with, and in several cases surpassing, prior state-of-the-art larger models across multiple benchmarks, as shown in Figure 2, while substantially reducing training cost. For example, compared with Z-Image (6B) (Cai et al., 2025a), *Lens* (3.8B) attains competitive or superior results while using only approximately 19.3% of its training compute. Specifically, *Lens* requires 192K A100 GPU hours (312 TFLOPS, BF16), whereas Z-Image requires 314K H800 GPU hours (989.5 TFLOPS, BF16).[1] Moreover, due to its smaller model size, *Lens* also enables faster inference under the same number of denoising steps.

Despite its reduced model size, the high training efficiency and strong performance of *Lens* are largely attributed to two additional factors: *data information density per training batch* and *convergence speed*.

Data Information Density per Training Batch. Given a training batch consisting of a set of *image-text pairs*, our objective is to maximize the amount of useful visual-semantic supervision contained in each optimization step. To this end, we increase information density from both the text and image perspectives:

- **Text Information Density.** Conventional short captions provide limited supervision, as they often describe only the most salient object or scene category. In contrast, dense captions encode richer semantic details, including objects, attributes, spatial relationships, actions, and background context, allowing each image-text pair to provide stronger training signals. This effectively increases the text-side information density of the dataset. Accordingly, *Lens* is trained on 800M densely captioned

[1]This comparison uses peak BF16 TFLOPS to normalize GPU types. Actual efficiency may differ due to memory bandwidth, MFU, and communication overhead. Re-captioning costs are excluded, as this one-time preprocessing can be reused for future models.

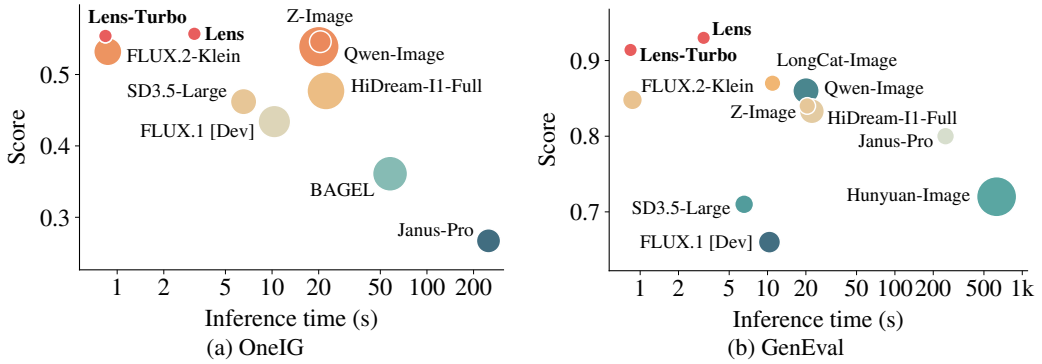


Figure 2: Comparison of inference time and benchmark performance on OneIG (Chang et al., 2025) and GenEval (Ghosh et al., 2023) across representative T2I models. The x-axis denotes inference time on a single NVIDIA H100 GPU, the y-axis denotes the benchmark score, and the marker area is proportional to model size.

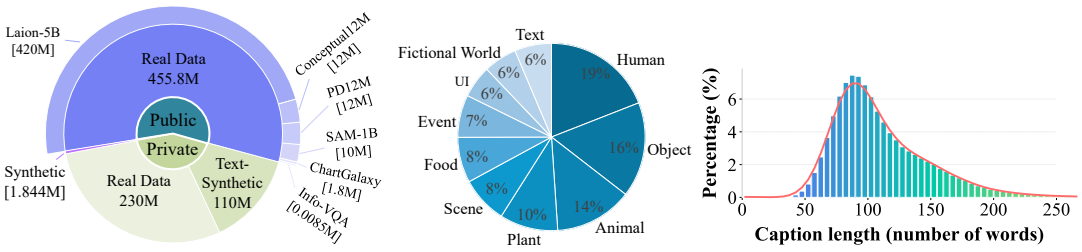
image-text pairs, where each caption is generated by a strong vision-language model, *GPT-4.1*, with an average length of 109 words.

- **Image Information Density.** We increase image-side information density by constructing each training batch from images with multiple resolutions (i.e., $\{512^2, 768^2, 1024^2\}$) and diverse aspect ratios (e.g., 1:2, 9:16, 1:1, and 4:3). This strategy significantly increases image information density within each training batch: multi-resolution training allows the model to learn visual content at different levels of detail, from global scene structure to fine local patterns, while multi-aspect-ratio training exposes it to diverse object arrangements, spatial relationships, and compositional layouts. Moreover, a useful by-product of this strategy is strong *resolution and aspect-ratio generalization* at inference time: the model generalizes well to unseen aspect ratios (e.g., 5:4 and 6:7) and to resolutions up to 1440^2. This capability removes the need for costly high-resolution training, which further enhances overall training efficiency when high-resolution generation is desired.

Convergence Speed. *Lens* further enhances training efficiency by accelerating optimization convergence. We explore several architectural design choices that allow the model to learn more effectively and reach strong performance with fewer training iterations. These studies include:

- **VAE Variants.** We systematically study different VAE variants, including conventional VAEs used in FLUX.1 (Black Forest Labs, 2024) and SD3 (Esser et al., 2024), as well as semantic VAEs adopted in FLUX.2 (Labs, 2025) and VTP (Yao et al., 2025a). Instead of relying on proxy metrics such as rFID or class-conditional ImageNet generation, we directly evaluate each VAE within the T2I pipeline using a 130M subset of our training data.
- **Language Encoder Variants.** The language encoder provides text-conditioning features for diffusion modeling. We find that stronger language encoders not only accelerate optimization convergence but also improve multilingual generalization. Specifically, although the model is trained only on English image-text pairs, a strong language encoder enables robust inference-time generalization to other languages, such as Chinese and French. This multilingual generalization substantially reduces data requirements and training costs in scenarios where the model needs to handle multilingual inputs. Based on careful ablation studies, we adopt GPT-OSS (OpenAI, 2025) as the language encoder.

After efficient pre-training, *Lens* generates diverse images, but their aesthetic quality may vary and some outputs may contain artifacts. We apply reinforcement learning (RL) as a post-training step to suppress artifacts, improve visual composition, and enforce consistency with real-world physical rules. A key finding is that RL data must be sufficiently diverse and cover the original training distribution to avoid performance degradation on certain input types. To this end, we construct the *Lens-RL-8K* prompt set with taxonomy-driven coverage of diverse generation scenarios. Experiments show that post-training on *Lens-RL-8K* significantly improves generation performance across a broad range of scenarios.



(a) Lens-800M. (b) Lens-RL-8K. (c) Lens-800M.

Figure 3: Distribution of (a) the pre-training dataset, *Lens-800M*, (b) the RL dataset, *Lens-RL-8K*, and (c) caption length distribution of the *Lens-800M* dataset, caption length is measured by the number of words, with an average length of approximately 109 words.

Additionally, following modern T2I systems, we equip *Lens* with a *reasoner* module that can be instantiated with different LLMs. The reasoner converts ambiguous or underspecified user requests into detailed prompts aligned with the training-caption distribution. It takes the user request and a system prompt as input, where the system prompt specifies guidelines for constructing suitable T2I prompts. We further introduce a *training-free system prompt search strategy* to optimize these guidelines, enabling the reasoner to generate prompts that better align with the T2I model. Note that reasoner-based prompt rewriting is now a standard practice in modern T2I systems; to ensure a fair comparison, we report results both with and without the reasoner in our experiments.

Overall, in this paper we systematically investigate a set of training efficiency factors that are often overlooked in practice, including data captioning strategies, VAE selection criteria, language encoder choices, and training-data composition for RL-based post-training. For each factor, we provide controlled ablation studies with quantitative analysis, yielding actionable insights for building T2I foundation models. Importantly, these strategies are complementary to conventional training acceleration approaches, such as architectural innovations and distributed-system optimization. Guided by these findings, *Lens* achieves performance competitive with larger state-of-the-art models at substantially lower training cost. Its compact model size also enables faster inference: by default, *Lens* generates a 1024^2 image in **3.15** seconds on a single NVIDIA H100 GPU using 20 denoising steps, while *Lens-Turbo*, a 4-step distilled variant, further reduces the generation time to **0.84** seconds.

2 METHOD

In this section, we present the details of *Lens*. We first describe the construction of the training dataset, *Lens-800M*, in Section 2.1. We then present the model architecture in Section 2.2, followed by the pre-training recipe in Section 2.3. In Section 2.4, we introduce our RL-driven post-training strategy, which is built on the carefully designed *Lens-RL-8K* dataset and optimized reward rubrics. We further introduce few-step distillation to distill *Lens* into *Lens-Turbo*, a 4-step generator that does not require CFG. Finally, Section 2.5 discusses inference configuration and training-free system-prompt search.

2.1 PRE-TRAINING DATA: LENS-800M

Data Distribution. Our pre-training corpus is constructed from four complementary sources to ensure content diversity: *(1) public real-world data*, including LAION-5B (Schuhmann et al., 2022), Conceptual12M (Changpinyo et al., 2021), PD12M (Meyer et al., 2024), SAM-1B (Kirillov et al., 2023), Infographic-VQA (Mathew et al., 2022), and ChartGalaxy (Li et al., 2026); *(2) public synthetic data*, including Flux-Reason6M (Fang et al., 2025) and Echo-4o (Ye et al., 2025); *(3) private data*, covering text-heavy visual content such as posters, slides, graphic designs, and general-domain images; and *(4) text synthetic data*, where text is rendered onto randomly sampled backgrounds with augmentations in blur, color, font, scale, and rotation to increase typographic and layout diversity. After the data filtering process described in Appendix D.1, the final pre-training dataset contains approximately *800M* high-quality images. The detailed data distribution is illustrated in Figure 3(a). We refer to this pre-training dataset as **Lens-800M**.

Captioning Images with Detailed Captions. For each image in *Lens-800M*, we employ a strong vision-language model, GPT-4.1 in our implementation, to generate a detailed, long-form *English* caption using the prompt described in Appendix F.1. At the same time, to preserve multilingual

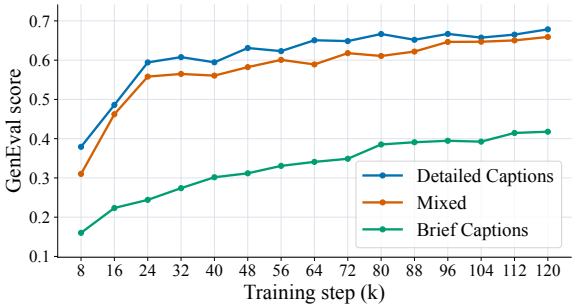


Figure 4: Caption-length ablation study.

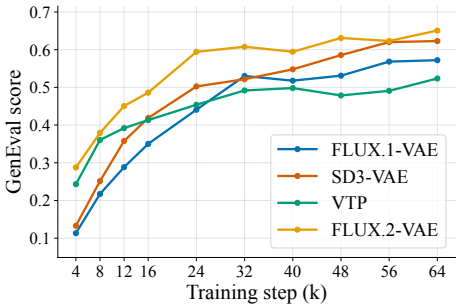


Figure 5: Ablation study on VAE variants.

rendering capabilities, any text appearing in the image is kept in its original language in the caption. Figure 3(c) presents the caption length statistics. Training examples are provided in Appendix C.2.

This design is motivated by three considerations. (1) *Improving data quality.* Web-crawled alt-text captions are often short, underspecified, and sometimes incorrect. Such noisy supervision forces the model to resolve ambiguity during training, leading to inefficient capacity usage and degraded learning signals (Betker et al., 2023; Chen et al., 2024). (2) *Bridging the training–inference gap.* In real-world usage, users frequently provide long and compositional prompts to describe desired images. Training on detailed captions better aligns the model with this inference-time distribution. (3) *Enhancing data efficiency.* Empirically, we observe that training *exclusively* on dense captions yields the best generation performance, outperforming short-caption training.

Ablation Study: Detailed vs. Brief Captions. To validate observation (3), we conduct a controlled ablation study. We randomly sample 130M images from the *Lens-800M* dataset to construct an ablation subset, denoted as **Lens-130M**. We train three small text-to-image models (referred to as **Lens-Toy**) with identical architectures (described in Section 2.2), each using a 1.2B-parameter image generation backbone and a Qwen3-0.6B text encoder. The only difference lies in the captioning strategy: *(i) Brief*: where GPT-4.1 generates short and sparse captions (e.g., "a photo of a cat") for each image in *Lens-130M*; *(ii) Detailed*, which uses our generated dense captions; and *(iii) Mixed*, a 50/50 combination of Brief and Detailed captions. We evaluate generation performance on the GenEval (Ghosh et al., 2023) benchmark. As shown in Figure 4, training with dense captions achieves better generation quality than the other variants, owing to improved data utilization efficiency.

> *Dense caption supervision improves the data information density of each training batch, leading to better data utilization and higher training efficiency.*

2.2 Architecture

Our model mainly consists of: (1) a *VAE* that encodes images into compact latents; (2) a *Latent Diffusion Transformer* that denoises text-conditioned image latents; (3) a *Reasoner* that converts ambiguous user requests into detailed, well-formed prompts.

VAE. We examine both classical VAEs, including those used in FLUX.1 (Black Forest Labs, 2024) and SD3 (Esser et al., 2024), and semantic VAEs, including those used in FLUX.2 (Labs, 2025) and VTP (Yao et al., 2025a). We do not use rFID to evaluate VAE performance, since reconstruction fidelity mainly measures how well a VAE reproduces a given image, rather than how effectively its latent space supports generative learning. We also avoid relying on class-conditional ImageNet generation as a proxy evaluation. Instead, we directly assess each VAE in the text-to-image generation setting by training *Lens-Toy* models on the *Lens-130M* dataset introduced in Section 2.1. As shown in Figure 5, FLUX.2's VAE achieves the best generation performance while also accelerating model convergence, and is therefore adopted as the VAE in *Lens*.

> *The VAE is crucial for both generation quality and model convergence. A strong VAE defines a more compact and semantically meaningful visual latent space, making text-image alignment easier to learn and reducing the number of optimization steps required for convergence.*

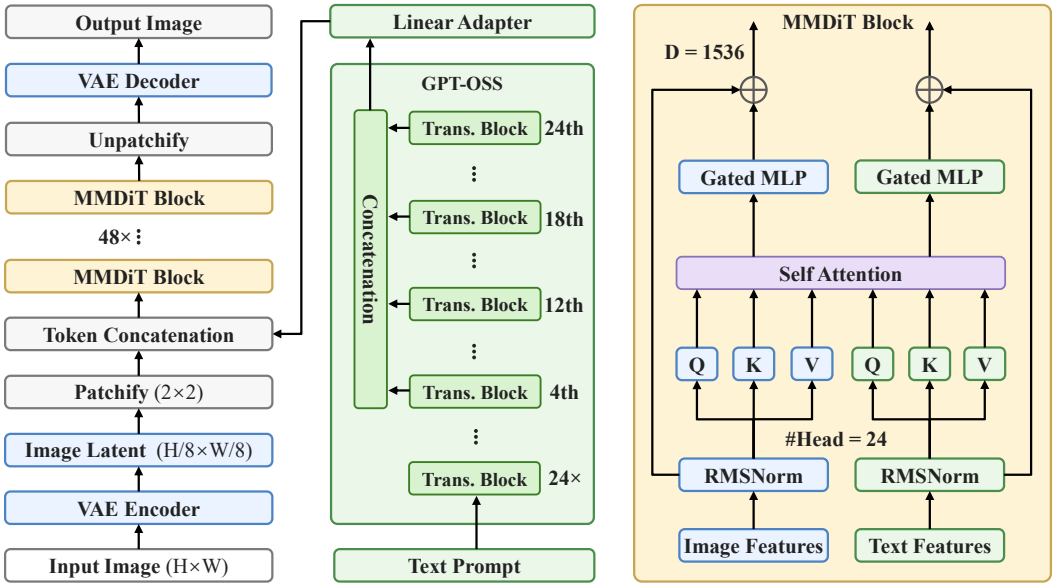


Figure 6: Architecture of the latent diffusion Transformer in *Lens* (left) and the detailed design of an MMDiT block (right). "Trans.": Transformer.

Latent Diffusion Transformer. We adopt an MMDiT-style (Esser et al., 2024) architecture for *Lens*, as illustrated in Figure 6. *Lens* is formulated as a latent diffusion model and trained with the standard flow-matching (Lipman et al., 2023) objective. Image latents are extracted using the FLUX.2 VAE, while text features are obtained from GPT-OSS (OpenAI, 2025), a 20B-parameter MoE language model with 3B activated parameters and 24 layers in total. To better leverage multi-level semantic representations, we extract GPT-OSS features from the 4th, 12th, 18th, and 24th layers and concatenate them along the feature dimension. A linear adapter is then applied to project the concatenated text representation into the same dimensionality as the image latents. The denoising backbone consists of 48 MMDiT blocks. Each block takes as input the concatenation of noisy image features and the text features produced by the previous MMDiT block, and processes them through two separate branches for image and text modalities. We use RMSNorm (Zhang & Sennrich, 2019) as the normalization layer and apply RoPE (Su et al., 2024) to the image features.

Ablation Study: Language Encoder. We consider two key factors when selecting the language encoder: (1) whether a stronger language encoder can facilitate text-image alignment, leading to better generation performance and faster convergence; and (2) whether it can enable multilingual generalization, i.e., training on English-only image-text pairs while supporting inference in other languages. To verify these effects, we compare four language encoders: GPT-OSS (MoE, 20B-A3B) (OpenAI, 2025) and Qwen3 (Yang et al., 2025) with different model sizes, including 0.6B, 1.7B, and 4B. We use *Lens-Toy* as the ablation model and construct four variants that differ only in the choice of language encoder. All variants are trained on *Lens-130M*, which contains 130M image-text pairs with English-only captions. Figures 7 and 8 present performance curves on the GenEval (Ghosh et al., 2023) benchmark as a function of training iterations. Based on these results, we adopt GPT-OSS as our language encoder.

> *Strong language encoders provide a richer semantic text space for text-image alignment, bringing three key benefits: improved prompt-following fidelity, faster model convergence, and, most importantly, support for multilingual inputs beyond the training language without additional multilingual image-text training data.*

Reasoner. The Reasoner is an independent language module placed before the T2I model. Its role is to interpret the user's raw input, refine ambiguous or underspecified instructions, and convert them into more detailed, coherent prompts optimized for generation. Because it functions independently of the T2I model's internal text encoder, the Reasoner can be easily swapped out without retraining the generation backbone. While we use GPT-5.5 as our default, the Reasoner is compatible with

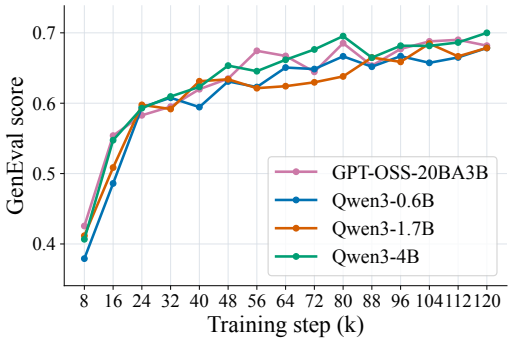


Figure 7: Study of different language encoders for English text-conditioned generation.

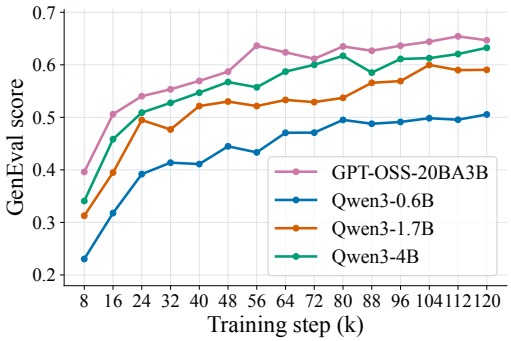


Figure 8: Study of language encoders for multilingual text-conditioned generation, averaged over five common languages (EN/ZH/FR/JA/ES).

various commercial and open-source LLMs. Our evaluations in Appendix B.2 show that even using an open-source model like GPT-OSS provides substantial gains. This setup is particularly efficient: since GPT-OSS already functions as our text encoder, employing it as the Reasoner adds zero extra GPU memory cost, demonstrating that our framework can achieve superior results without relying on costly commercial APIs.

2.3 Pre-training

Low-resolution Pre-training. We first pre-train *Lens* at a fixed resolution of 512×512 on 128 NVIDIA A100 80GB GPUs for 400K iterations. Details are provided in Appendix D.2.

Mixed-resolution Continual Training. Starting from the low-resolution checkpoint, we continue training *Lens* for another 400K iterations over mixed resolutions on 128 NVIDIA A100 80GB GPUs. Specifically, we construct resolution buckets from three base image areas, 512^2, 768^2, and 1024^2, combined with nine aspect ratios: 1:2, 9:16, 2:3, 3:4, 1:1, 4:3, 3:2, 16:9, and 2:1. This yields 27 concrete resolution buckets. More training details are provided in Appendix D.2.

Resolution and Aspect-ratio Generalization after Pre-training. Although the base model is trained on only 27 resolutions, constructed from 3 base areas and 9 aspect ratios, it generalizes well to unseen resolutions and aspect ratios at inference time. Specifically, it can generate images with arbitrary aspect ratios ranging from 1:2 to 2:1 and image areas up to 1440^2, even though training does not include resolutions between 1024^2 and 1440^2, nor aspect ratios outside the predefined bucket set. This suggests that mixed-resolution pre-training does not simply encourage the model to memorize a fixed set of resolution buckets. Instead, exposure to diverse spatial scales and aspect ratios enables the model to learn more continuous and resolution-aware image representations. Moreover, the use of RoPE-based positional encoding may further facilitate such generalization, as it represents positions in a relative and extrapolatable manner.

> *Mixed-resolution training enables strong resolution generalization at inference time, allowing the model to generate images at unseen sizes and diverse aspect ratios. This reduces the need to collect and train on images covering every target resolution and aspect-ratio combination. More importantly, since training on higher-resolution images increases computational cost quadratically with image size, resolution generalization allows the model to produce higher-resolution outputs while being trained primarily on lower-resolution images. This substantially reduces training cost and improves overall training efficiency.*

2.4 Post-training

After pre-training, our base model, *Lens-Base*, can strictly follow user prompts and generate diverse images. However, the generated images may still contain visual artifacts. To further improve generation quality and reliability, we adopt reinforcement learning as a post-training strategy.

Lens-RL-8K Dataset. *Lens-RL-8K* is a prompt dataset designed for RL-based post-training, consisting of 8,406 prompts that cover a broad range of T2I generation scenarios. A key observation in this

Table 1: Left: Comparison of different *Lens-RL* variants trained on different subsets of *Lens-RL-8K* on GenEval. Right: Comparison between *Lens-RL* models trained on the full *Lens-RL-8K* set and a subset excluding text prompts on two text-rendering benchmarks, CVTG and OneIG (EN).

RL Training Set	GenEval
1/4 Full set	0.916
1/2 Full set	0.920
Full set	0.930

RL Training Set	CVTG			OneIG (EN)
	Avg.	NED	CLIP	Text
Full set *w/o* text	0.832	0.928	0.795	0.946
Full set	0.869	0.951	0.814	0.960

work is that RL prompts should match the generation-scenario distribution of the pre-training data as comprehensively as possible. This enables post-training to improve the model's overall generation quality and alignment across diverse scenarios, rather than overfitting to a narrow set of prompt types. The detailed taxonomy-driven construction pipeline of *Lens-RL-8K* is described in Appendix D.3. The category distribution of the prompts is shown in Figure 3(b).

Rubric Generation. Rubrics define the key aspects used to assess images generated by *Lens*. Given a prompt $\mathcal{P}$ from the *Lens-RL-8K* dataset, we first generate 10 sample-aware rubrics using GPT-4.1. Specifically, we feed $\mathcal{P}$ together with the system prompt described in Appendix F.2 into GPT-4.1 to produce prompt-specific rubrics. We further append a global rubric, i.e., "*Verify that the entire image is structurally coherent and physically plausible*". As a result, each prompt is associated with a set of evaluation rubrics. We also provide several prompt-rubric examples in Appendix C.3.

DiffusionNFT with VLM as Reward Function. We adopt DiffusionNFT (Zheng et al., 2025) to optimize *Lens-Base*, using GPT-4.1-mini as the reward function. Specifically, at each optimization step, we randomly sample 48 prompt-rubric pairs and generate 24 images at different resolutions for each prompt using the current policy model. We then feed each generated image, together with its corresponding rubrics, into GPT-4.1-mini. Guided by the system prompt described in Appendix F.3, GPT-4.1-mini produces rewards that are used by DiffusionNFT to optimize the policy model. We train the RL policy for 180 steps on 64 NVIDIA A100 GPUs. Further details are provided in Appendix D.4.

Ablation Study: Various RL Datasets. To verify that prompt diversity in the RL dataset is crucial for post-training performance, we conduct two ablation studies. First, we compare our default model, *Lens-RL*, trained on the full *Lens-RL-8K* dataset, with a variant trained on *Lens-RL-8K* after removing text-related prompts. Second, we compare *Lens-RL* with variants trained on smaller subsets of *Lens-RL-8K*, including one-half and one-quarter of the full dataset, which substantially reduce the diversity of RL training prompts. All models are initialized from the same base model and trained for the same 180 RL steps. The results are reported in Table 1.

> *A broad and well-balanced RL prompt set enables the model to improve across diverse generation scenarios, whereas reduced or biased prompt coverage limits generalization.*

Few-step Distillation. To improve sampling efficiency, we distill *Lens-RL* into *Lens-Turbo*, a 4-step generator distilled from a curated, well-balanced image-caption dataset. Our recipe combines effective techniques from DMD2 (Yin et al., 2024a), decoupled-DMD (Liu et al., 2025a), and SenseFlow (Ge et al., 2025), together with R1 regularization (Roth et al., 2017; Lin et al., 2025) on the adversarial loss to improve training stability. *Lens-Turbo* largely preserves the image quality and prompt-following ability of the original model. Details are provided in Appendix D.5.

2.5 Inference

By default, we first use a reasoner to refine ambiguous or underspecified user inputs into more detailed prompts. The refined prompts are then fed into *Lens* for **20**-step generation with CFG set to 5.0. For faster inference, *Lens-Turbo* performs **4**-step generation without CFG.

Training-free System-prompt Search. This technique aims to find a better system prompt for the reasoner, enabling it to more effectively convert user requests into suitable T2I model prompts. It requires no model training. Instead, we iteratively feed the previous system prompt and generation analysis, i.e., textual summaries of failure cases, into GPT-5.5, and ask it to rewrite the system prompt. We find that this training-free strategy produces improved system prompts for the reasoner. This strategy is not specific to our T2I system; it can also be applied to other T2I models.

Table 2: Comparison of *Lens* with 20-step inference and *Lens-Turbo* with 4-step inference against state-of-the-art models across four benchmarks. We report overall scores for *OneIG*, *GenEval*, and *LongText (EN)*, and average score, normalized edit distance (NED), and CLIP score for *CVTG*. Detailed benchmark comparisons are provided in Tables 3, 4, and 5 in Appendix B.1. The best open-source results are highlighted in **bold**, and the second-best results are underlined.

Model	Size	OneIG		GenEval	LongText (EN)	CVTG		
		EN	ZH			Avg.	NED	CLIP
Commercial models								
Kolors 2.0	–	0.434	0.426	–	0.258	–	–	–
Seedream 3.0	–	0.530	0.528	0.843	0.896	0.592	0.854	0.782
Seedream 4.0	–	0.573	0.554	0.840	0.921	0.892	0.951	0.785
GPT Image 1 [High]	–	0.533	0.474	0.840	0.956	0.857	0.948	0.798
Nano Banana 2.0	–	0.578	0.567	–	0.981	–	–	–
Open-source models								
Janus-Pro	7B	0.267	0.240	0.800	0.019	–	–	–
BAGEL	14B	0.361	0.370	–	0.373	–	–	–
HiDream-I1-Full	17B	0.477	0.337	0.833	0.543	–	–	–
SD3.5 Large	8B	0.462	–	0.710	–	0.655	0.847	0.780
FLUX.1 [Dev]	12B	0.434	–	0.660	0.607	0.496	0.688	0.740
FLUX.2-Klein	9B	0.532	0.430	0.848	0.864	–	–	–
Z-Image-Turbo	6B	0.528	0.507	0.823	0.917	0.859	0.928	0.805
Z-Image	6B	0.546	0.535	0.840	0.935	0.867	0.937	0.797
Qwen-Image	20B	0.539	**0.548**	0.868	**0.943**	0.829	0.930	0.806
Hunyuan-Image-3.0	80B	–	–	0.720	–	0.765	0.877	0.812
LongCat-Image	6B	–	–	0.870	–	0.866	0.936	0.786
Lens-Turbo (4-step)	**3.8B**	0.554	0.519	0.914	0.927	**0.889**	**0.965**	**0.815**
Lens (20-step)	**3.8B**	**0.557**	0.525	**0.930**	0.937	0.869	0.951	0.814

3 Comparison with State-of-the-art Models

Main Results. In Table 2, we evaluate *Lens* against state-of-the-art models on four text-to-image benchmarks: **OneIG** (Chang et al., 2025), **GenEval** (Ghosh et al., 2023), **LongText (EN)** (Geng et al., 2025), and **CVTG (EN)** (Du et al., 2025). These benchmarks cover general prompt following, compositional grounding, and visual text rendering. Details of each benchmark are provided in Appendix B.1. With a 3.8B-parameter backbone, *Lens* achieves the highest OneIG (EN) and GenEval scores among the compared open-source models, at 0.557 and 0.930, respectively, while remaining competitive on text rendering. With only four denoising steps, *Lens-Turbo* retains strong performance and achieves the best open-source results across all three reported CVTG metrics.

Visualizations. Appendix C.1 presents *Lens-Turbo* outputs covering general generation, text rendering, portrait generation, and multilingual prompt following. These examples use an image area of 1440^2 pixels with varying aspect ratios, illustrating high-resolution generation beyond the training resolutions.

4 Conclusion

In this work, we introduce **Lens**, a 3.8B-parameter foundational T2I model designed for training-time efficiency. By improving data information density through dense captions and mixed-resolution/aspect-ratio pre-training on the proposed *Lens-800M* dataset, and by accelerating convergence with carefully selected VAE and language encoder designs, *Lens* achieves strong generation quality at substantially lower training cost. We further enhance the model through RL-based post-training on the diverse *Lens-RL-8K* prompt set, together with system-level optimizations including a reasoner, training-free system-prompt search, and few-step distillation. Extensive experiments show that *Lens* achieves performance competitive with, and in several cases surpassing, larger state-of-the-art models across multiple benchmarks while enabling fast inference, demonstrating that efficient training strategies can substantially improve the scalability of foundational T2I models.

AI USE STATEMENT

We use language and vision-language models for image captioning, RL prompt and rubric construction, reward evaluation, and inference-time prompt refinement, including system-prompt search. These uses are described in the method. We also use language models to improve the clarity and readability of the manuscript. The research ideas and methodological design were developed by the authors without generative AI assistance. The authors take responsibility for the final content and claims of this paper.

ETHICS STATEMENT

Our training corpus combines public datasets, private image collections, and synthetic data, with a multi-stage filtering pipeline that includes NSFW content filtering. We encourage responsible use of our model that respects privacy and intellectual property rights. All authors have read and adhere to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

We describe the data construction, model architecture, training procedures, and inference settings in the main text, with implementation details, prompts, and benchmark descriptions provided in the appendices. We will release both the pre-trained and post-trained models to support reproducibility and further research.

REFERENCES

Alan Baade, Eric Ryan Chan, Kyle Sargent, Changan Chen, Justin Johnson, Ehsan Adeli, and Li Fei-Fei. Latent forcing: Reordering the diffusion trajectory for pixel-space image generation. *arXiv preprint arXiv:2602.11401*, 2026.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. Improving image generation with better captions. Technical report, OpenAI, 2023.

Black Forest Labs. FLUX: Open-weight text-to-image models. `https://github.com/black-forest-labs/flux`, 2024.

Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Xin Jin, Liangchen Li, Zhen Li, Zhong-Yu Li, David Liu, Dongyang Liu, Junhan Shi, Qilong Wu, Feng Yu, Chi Zhang, Shifeng Zhang, and Shilin Zhou. Z-Image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025a.

Qi Cai, Jingwen Chen, Yang Chen, Yehao Li, Fuchen Long, Yingwei Pan, Zhaofan Qiu, Yiheng Zhang, Fengbin Gao, Peihan Xu, et al. Hidream-i1: A high-efficient image generative foundation model with sparse diffusion transformer. *arXiv preprint arXiv:2505.22705*, 2025b.

Siyu Cao, Hangting Chen, Peng Chen, Yiji Cheng, Yutao Cui, Xinchi Deng, Ying Dong, Kipper Gong, Tianpeng Gu, Xiusen Gu, et al. Hunyuanimage 3.0 technical report. *arXiv preprint arXiv:2509.23951*, 2025.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. pp. 9650–9660, 2021.

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. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. ShareGPT4V: Improving large multi-modal models with better captions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. 2024.

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

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 *Forty-first international conference on machine learning*, 2024.

Rongyao Fang, Aldrich Yu, Chengqi Duan, Linjiang Huang, Shuai Bai, Yuxuan Cai, Kun Wang, Si Liu, Xihui Liu, and Hongsheng Li. Flux-reason-6m & prism-bench: A million-scale text-to-image reasoning dataset and comprehensive benchmark. *arXiv preprint arXiv:2509.09680*, 2025.

Xuelu Feng, Yunsheng Li, Ziyu Wan, Zixuan Gao, Junsong Yuan, Dongdong Chen, and Chunming Qiao. Rubricrl: Simple generalizable rewards for text-to-image generation. *arXiv preprint arXiv:2511.20651*, 2025.

Xingtong Ge, Xin Zhang, Tongda Xu, Yi Zhang, Xinjie Zhang, Yan Wang, and Jun Zhang. Senseflow: Scaling distribution matching for flow-based text-to-image distillation. *arXiv preprint arXiv:2506.00523*, 2025.

Zigang Geng, Yibing Wang, Yeyao Ma, Chen Li, Yongming Rao, Shuyang Gu, Zhao Zhong, Qinglin Lu, Han Hu, Xiaosong Zhang, Linus, Di Wang, and Jie Jiang. X-omni: Reinforcement learning makes discrete autoregressive image generative models great again. *arXiv preprint arXiv:2507.22058*, 2025.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Google. Introducing gemini 2.5 flash image, our state-of-the-art image model. `https://developers.googleblog.com/en/introducing-gemini-2-5-flash-image/`, 2025. Also known as Nano Banana.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*, 2025.

Yun He, Wenzhe Li, Hejia Zhang, Songlin Li, Karishma Mandyam, Sopan Khosla, Yuanhao Xiong, Nanshu Wang, Xiaoliang Peng, Beibin Li, et al. Advancedif: Rubric-based benchmarking and reinforcement learning for advancing llm instruction following. *arXiv preprint arXiv:2511.10507*, 2025.

Jonathan Heek, Emiel Hoogeboom, Thomas Mensink, and Tim Salimans. Unified latents (ul): How to train your latents. *arXiv preprint arXiv:2602.17270*, 2026.

Jiwoo Hong, Sayak Paul, Noah Lee, Kashif Rasul, James Thorne, and Jongheon Jeong. Margin-aware preference optimization for aligning diffusion models without reference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4744–4752, 2026.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, et al. Reinforcement learning with rubric anchors. *arXiv preprint arXiv:2508.12790*, 2025.

Dengyang Jiang, Dongyang Liu, Zanyi Wang, Qilong Wu, Liuzhuozheng Li, Hengzhuang Li, Xin Jin, David Liu, Zhen Li, Bo Zhang, et al. Distribution matching distillation meets reinforcement learning. *arXiv preprint arXiv:2511.13649*, 2025.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

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.

Kuaishou Kolors Team. Kolors 2.0. https://klingai.com/app.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18262–18272, 2025.

Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Yiming Cheng, Miles Yang, Zhao Zhong, and Liefeng Bo. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025.

Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Yusuke Kato, and Kazuki Kozuka. Aligning diffusion models by optimizing human utility. *Advances in Neural Information Processing Systems*, 37:24897–24925, 2024.

Zhen Li, Duan Li, Yukai Guo, Xinyuan Guo, Bowen Li, Lanxi Xiao, Shenyu Qiao, Jiashu Chen, Zijian Wu, Hui Zhang, Xinhuan Shu, and Shixia Liu. Chartgalaxy: A dataset for infographic chart understanding and generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=P4lFbvZ4HH.

Zhanhao Liang, Yuhui Yuan, Shuyang Gu, Bohan Chen, Tiankai Hang, Mingxi Cheng, Ji Li, and Liang Zheng. Aesthetic post-training diffusion models from generic preferences with step-by-step preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13199–13208, 2025.

Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion adversarial post-training for one-step video generation. *arXiv preprint arXiv:2501.08316*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

Dongyang Liu, Peng Gao, David Liu, Ruoyi Du, Zhen Li, Qilong Wu, Xin Jin, Sihan Cao, Shifeng Zhang, Hongsheng Li, and Steven Hoi. Decoupled dmd: Cfg augmentation as the spear, distribution matching as the shield. *arXiv preprint arXiv:2511.22677*, 2025a.

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*, 2025b.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *ICLR*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*, 2022.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1697–1706, 2022.

Jordan Meyer, Nick Padgett, Cullen Miller, and Laura Exline. Public domain 12m: A highly aesthetic image-text dataset with novel governance mechanisms, 2024. URL https://arxiv.org/abs/2410.23144.

OpenAI. Introducing our latest image generation model in the api. https://openai.com/index/image-generation-api/, 2025. Official announcement of gpt-image-1.

OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

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

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

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. pp. 8748–8763, 2021.

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 (CVPR)*, 2022.

Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. *arXiv preprint arXiv:1705.09367*, 2017.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, October 2020. URL https://arxiv.org/abs/2010.02502.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Stability AI. Stable Diffusion 3.5. `https://stability.ai/news-updates/introducing-stable-diffusion-3-5`, 2024. Official model release announcement.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Meituan LongCat Team, Hanghang Ma, Haoxian Tan, Jiale Huang, Junqiang Wu, Jun-Yan He, Lishuai Gao, Songlin Xiao, Xiaoming Wei, Xiaoqi Ma, et al. Longcat-image technical report. *arXiv preprint arXiv:2512.07584*, 2025.

Team Seedream, Yunpeng Chen, Yu Gao, Lixue Gong, Meng Guo, Qiushan Guo, Zhiyao Guo, Xiaoxia Hou, Weilin Huang, Yixuan Huang, et al. Seedream 4.0: Toward next-generation multimodal image generation. *arXiv preprint arXiv:2509.20427*, 2025.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Yibin Wang, Zhimin Li, Yuhang Zang, Yujie Zhou, Jiazi Bu, Chunyu Wang, Qinglin Lu, Cheng Jin, and Jiaqi Wang. Pref-grpo: Pairwise preference reward-based grpo for stable text-to-image reinforcement learning. *arXiv preprint arXiv:2508.20751*, 2025.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. Sana: Efficient high-resolution text-to-image synthesis with linear diffusion transformers. In *International Conference on Learning Representations*, 2025.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning rl with pretraining in diffusion models. *arXiv preprint arXiv:2509.25050*, 2025a.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025b.

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.

Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8941–8951, 2024a.

Shentao Yang, Tianqi Chen, and Mingyuan Zhou. A dense reward view on aligning text-to-image diffusion with preference. *arXiv preprint arXiv:2402.08265*, 2024b.

Jingfeng Yao, Yuda Song, Yucong Zhou, and Xinggang Wang. Towards scalable pre-training of visual tokenizers for generation. *arXiv preprint arXiv:2512.13687*, 2025a.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. pp. 15703–15712, 2025b.

Junyan Ye, Dongzhi Jiang, Zihao Wang, Leqi Zhu, Zhenghao Hu, Zilong Huang, Jun He, Zhiyuan Yan, Jinghua Yu, Hongsheng Li, et al. Echo-4o: Harnessing the power of gpt-4o synthetic images for improved image generation. *arXiv preprint arXiv:2508.09987*, 2025.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024b.

Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. *Advances in Neural Information Processing Systems*, 37:128940–128966, 2024.

Huizhuo Yuan, Zixiang Chen, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning of diffusion models for text-to-image generation. *Advances in Neural Information Processing Systems*, 37: 73366–73398, 2024.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

Shilong Zhang, He Zhang, Zhifei Zhang, Chongjian Ge, Shuchen Xue, Shaoteng Liu, Mengwei Ren, Soo Ye Kim, Yuqian Zhou, Qing Liu, et al. Both semantics and reconstruction matter: Making representation encoders ready for text-to-image generation and editing. *arXiv preprint arXiv:2512.17909*, 2025.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *NeurIPS*, 2023.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. *arXiv preprint arXiv:2509.16117*, 2025.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. In *International Conference on Learning Representations*, 2025.

This appendix includes:

- Related work discussion (Appendix A).
- Benchmark details, additional results, and analyses (Appendix B).
- Generation visualizations, training-data visualizations, and rubric visualizations (Appendix C).
- Implementation details (Appendix D).
- Computational cost analysis (Appendix E).
- All prompts used in this work (Appendix F).
- Broader impacts and limitations of this work (Appendix G).

A Related Works

A.1 Foundational T2I Models

Text-to-image (T2I) foundation models have progressed from latent diffusion models to large-scale diffusion Transformers, rectified-flow models, and proprietary multimodal image-generation systems. Early latent diffusion models (LDMs), including Stable Diffusion, established the paradigm of performing generation in a compressed latent space, substantially reducing training and inference costs compared with pixel-space diffusion (Rombach et al., 2022). Building on this paradigm, SDXL further improves high-resolution synthesis through a larger UNet, stronger text conditioning, multi-aspect-ratio training, and a dedicated refinement stage (Podell et al., 2024).

More recent models have shifted toward Transformer-based backbones and rectified-flow objectives. Stable Diffusion 3 and its 3.5 variants adopt MMDiT-style architectures with rectified-flow training (Esser et al., 2024; Stability AI, 2024), following the broader trend of scalable diffusion Transformers (Peebles & Xie, 2023). Recent open-source and commercial systems, such as FLUX (Black Forest Labs, 2024), SANA (Xie et al., 2025), HiDream-I1 (Cai et al., 2025b), Qwen-Image (Wu et al., 2025), Hunyuan-Image-3.0 (Cao et al., 2025), Z-Image (Cai et al., 2025a), GPT Image (OpenAI, 2025), Gemini 2.5 Flash Image (Nano Banana) (Google, 2025), Kolors 2.0 (Kuaishou Kolors Team), and Seedream 4.0 (Team Seedream et al., 2025), further advance visual quality, prompt following, text rendering, image editing, reference consistency, and inference efficiency.

In parallel, unified multimodal and autoregressive generators aim to integrate image synthesis with language modeling and visual understanding. Janus-Pro (Chen et al., 2025) scales a unified autoregressive framework for both multimodal understanding and generation. Transfusion (Zhou et al., 2025) combines next-token prediction with diffusion over continuous image representations, while BAGEL (Deng et al., 2025) scales decoder-only multimodal pretraining on interleaved text, image, video, and web data. Although these approaches achieve strong performance and broaden the capabilities of T2I systems, they often require increasingly large models, massive training data, and complex system designs. This trend motivates our study of efficient foundation-model training, aiming to achieve competitive generation quality with substantially reduced computational cost.

A.2 Post-training for T2I Models

Post-training has become an important stage for improving the alignment and generation quality of T2I models. Early efforts mainly relied on supervised fine-tuning, while recent work has increasingly explored preference-based optimization and reinforcement learning (RL). These methods aim to improve model behavior beyond likelihood-based pre-training by directly optimizing for human preferences, reward-model judgments, or task-specific generation criteria.

One major line of work adapts direct preference optimization (DPO) to diffusion models. These methods train the model using positive-negative image pairs or preference sets, encouraging generations preferred by humans or reward models while suppressing less desirable outputs. Representative approaches include Diffusion-DPO (Wallace et al., 2024), D3PO (Yang et al., 2024a), SPO (Liang et al., 2025), and related variants (Yuan et al., 2024; Yang et al., 2024b; Li et al., 2024; Hong et al., 2026). Compared with online RL, DPO-style methods are relatively simple and stable, as they optimize preference objectives without requiring explicit reward maximization during sampling.

Table 3: Comparison of *Lens* with commercial and open-source models on **OneIG (EN)**. The best results are highlighted in **bold**, and the second-best results are underlined.

Model	Size	Alignment	Text	Reasoning	Style	Diversity	Overall
Commercial models							
Kolors 2.0	–	0.820	0.427	0.262	0.360	0.300	0.434
Seedream 3.0	–	0.818	0.865	0.275	0.413	0.277	0.530
Seedream 4.0	–	0.892	0.983	0.347	0.453	0.191	0.573
GPT Image 1 [High]	–	0.851	0.857	0.345	0.462	0.151	0.533
Nano Banana 2.0	–	0.888	0.944	0.334	0.481	0.245	0.578
Open-source models							
Janus-Pro	7B	0.553	0.001	0.139	0.276	**0.365**	0.267
BAGEL	14B	0.769	0.244	0.173	0.367	0.251	0.361
HiDream-I1-Full	17B	0.829	0.707	0.317	0.347	0.186	0.477
SD3.5 Large	8B	0.809	0.629	0.294	0.353	0.225	0.462
FLUX.1 [Dev]	12B	0.786	0.523	0.253	0.368	0.238	0.434
FLUX.2-Klein	9B	0.887	0.866	0.312	**0.442**	0.156	0.532
Z-Image-Turbo	6B	0.840	**0.994**	0.298	0.368	0.139	0.528
Z-Image	6B	0.881	0.987	0.280	0.387	0.194	0.546
Qwen-Image	20B	0.882	0.891	0.306	0.418	0.197	0.539
Lens-Turbo	**3.8B**	0.884	0.972	**0.349**	0.383	0.180	0.554
Lens	**3.8B**	**0.891**	0.960	0.343	0.404	0.186	**0.557**

Another line of work applies policy-gradient-based RL to the generation process. GRPO-based methods formulate image generation as a sequential decision-making problem and optimize the denoising or flow-matching trajectory directly. For flow-matching models, Flow-GRPO (Liu et al., 2025b) and its variants (Li et al., 2025; Wang et al., 2025) extend policy-gradient optimization to continuous generative dynamics by converting deterministic sampling into an equivalent stochastic formulation. Related methods, such as DiffusionNFT (Zheng et al., 2025) and AWM (Xue et al., 2025a), further investigate RL objectives over the generative process, introducing online optimization frameworks and advantage-aware or negative-aware objectives to guide model improvement.

Across both preference-optimization and RL-based post-training, reward design is a critical factor. A useful reward should capture not only global aesthetic quality, but also prompt faithfulness, object correctness, spatial and compositional consistency, text rendering, and safety-related constraints. Poorly designed rewards may cause reward hacking, reduced diversity, or misalignment between the optimization objective and human preference. Therefore, recent work (Feng et al., 2025; Gunjal et al., 2025; Huang et al., 2025; He et al., 2025) has increasingly emphasized fine-grained, multi-dimensional reward construction. This motivates our rubric-based post-training strategy, which provides explicit and structured evaluation criteria for RL optimization of T2I foundation models.

A.3 Distillation for T2I Models

A central challenge of diffusion- and flow-based T2I models is their expensive iterative sampling process. Early acceleration methods mainly improve the inference solver without modifying model parameters. Representative examples include deterministic samplers such as DDIM (Song et al., 2020), as well as high-order ODE solvers such as DPM-Solver (Lu et al., 2022) and UniPC (Zhao et al., 2023). Although these training-free methods can reduce the number of function evaluations, generation quality often degrades when the sampling budget becomes extremely small, especially for high-resolution text-conditioned synthesis under strong classifier-free guidance.

To achieve more aggressive acceleration, another line of work distills a multi-step teacher model into a one-step or few-step student. Progressive distillation (Salimans & Ho, 2022) iteratively halves the

Table 4: Comparison of *Lens* with commercial and open-source models on **GenEval**.

Model	Size	Single	Two	Count	Color	Position	Attribute	Overall
Commercial models								
Seedream 3.0	–	0.990	0.960	0.910	0.930	0.470	0.800	0.843
Seedream 4.0	–	0.990	0.920	0.720	0.910	0.760	0.740	0.840
GPT Image 1 [High]	–	0.990	0.920	0.850	0.920	0.750	0.610	0.840
Open-source models								
Janus-Pro	7B	0.990	0.890	0.590	0.900	0.790	0.660	0.800
HiDream-I1-Full	17B	**1.000**	**0.980**	0.790	0.910	0.600	0.720	0.833
SD3.5 Large	8B	0.980	0.890	0.730	0.830	0.340	0.470	0.710
FLUX.1 [Dev]	12B	0.980	0.810	0.740	0.790	0.220	0.450	0.660
FLUX.2-Klein	9B	0.931	0.957	0.828	0.915	0.718	0.740	0.848
Z-Image-Turbo	6B	**1.000**	0.950	0.770	0.890	0.650	0.680	0.823
Z-Image	6B	**1.000**	0.940	0.780	**0.930**	0.620	0.770	0.840
Qwen-Image	20B	0.990	0.920	<u>0.890</u>	0.880	0.760	0.770	0.868
Hunyuan-Image-3.0	80B	**1.000**	0.920	0.480	0.820	0.420	0.630	0.720
LongCat-Image	6B	0.990	**0.980**	0.860	0.860	0.750	0.730	0.870
Lens-Turbo	**3.8B**	<u>0.997</u>	0.947	**0.909**	0.891	<u>0.895</u>	<u>0.845</u>	<u>0.914</u>
Lens	**3.8B**	0.994	<u>0.970</u>	**0.909**	<u>0.923</u>	**0.915**	**0.868**	**0.930**

number of sampling steps by matching deterministic teacher trajectories. Consistency models (Song et al., 2023) learn a self-consistent mapping from noisy states to clean data, enabling fast generation with few denoising steps. Latent Consistency Models (Luo et al., 2023) extend this idea to latent-space T2I models, while InstaFlow (Liu et al., 2024) accelerates rectified-flow models by straightening probability-flow trajectories and improving noise-data couplings. These methods demonstrate the potential of trajectory- or consistency-based distillation, but their performance can still degrade under very small step budgets or complex text-conditioned generation.

Recent methods further improve few-step T2I distillation by introducing distribution-level and adversarial supervision. Adversarial Diffusion Distillation (Sauer et al., 2023) combines score distillation with an adversarial objective to improve perceptual quality. Distribution Matching Distillation (Yin et al., 2024b) directly matches the student distribution to the target data distribution using real and fake score estimates, reducing the reliance on paired teacher trajectories. Its improved variants, including DMD2 (Yin et al., 2024a), decoupled-DMD (Liu et al., 2025a), DMD-R (Jiang et al., 2025), and SenseFlow (Ge et al., 2025), further enhance training stability, guidance distillation, and distribution alignment for large-scale T2I models. Despite these advances, few-step distillation remains sensitive to teacher guidance, timestep design, fake-score estimation, and adversarial stability.

A.4 VAE

Variational Autoencoders (VAEs) (Kingma & Welling, 2013) are widely used as image tokenizers in diffusion-based generation models, serving as the bridge between pixel space and latent representations. Conventional tokenizers are typically trained with reconstruction-oriented objectives, aiming to preserve as much pixel-level information as possible. However, recent studies suggest that latents optimized purely for reconstruction are not necessarily optimal for generative modeling. Such latents may be semantically entangled, difficult for diffusion models to learn, and can slow down convergence during training. *Reconstruction vs. Generation* (Yao et al., 2025b) formally analyzes this conflict in latent diffusion models, while *Both Semantics and Reconstruction Matter* (Zhang et al., 2025) empirically shows that excessive reconstruction pressure tends to prioritize low-level details over semantic structure, which can hurt text-to-image generation and editing performance.

Motivated by this mismatch, recent work has explored generation-friendly tokenizers that better align the latent space with downstream generative objectives. REPA-E (Leng et al., 2025) aligns encoder

Table 5: Comparison of *Lens* with commercial and open-source models on **LongText (EN)** and **CVTG**.

Model	Size	LongText (EN)	CVTG						
			2R	3R	4R	5R	Avg.	NED	CLIP
Commercial models									
Kolors 2.0	–	0.258	–	–	–	–	–	–	–
Seedream 3.0	–	0.896	0.628	0.596	0.604	0.561	0.592	0.854	0.782
Seedream 4.0	–	0.921	0.890	0.915	0.899	0.887	0.892	0.951	0.785
GPT Image 1 [High]	–	0.956	0.878	0.866	0.873	0.822	0.857	0.948	0.798
Nano Banana 2.0	–	0.981	–	–	–	–	–	–	–
Open-source models									
Janus-Pro	7B	0.019	–	–	–	–	–	–	–
BAGEL	14B	0.373	–	–	–	–	–	–	–
HiDream-I1-Full	17B	0.543	–	–	–	–	–	–	–
SD3.5 Large	8B	–	0.729	0.683	0.657	0.594	0.655	0.847	0.780
FLUX.1 [Dev]	12B	0.607	0.609	0.553	0.466	0.432	0.496	0.688	0.740
FLUX.2-Klein	9B	0.864	–	–	–	–	–	–	–
Z-Image-Turbo	6B	0.917	0.887	0.866	0.863	0.835	0.859	0.928	0.805
Z-Image	6B	0.935	0.901	0.872	0.865	0.851	0.867	0.937	0.797
Qwen-Image	20B	**0.943**	0.837	0.836	0.831	0.816	0.829	0.930	0.806
Hunyuan-Image-3.0	80B	–	0.830	0.764	0.738	0.728	0.765	0.877	0.812
LongCat-Image	6B	–	**0.913**	0.874	0.856	0.831	0.866	0.936	0.786
Lens-Turbo	**3.8B**	0.927	0.893	**0.892**	**0.894**	**0.878**	**0.889**	**0.965**	**0.815**
Lens	**3.8B**	0.937	0.897	0.881	0.872	0.827	0.869	0.951	0.814

representations with diffusion Transformer features, thereby easing optimization for the generative model. Unified Latents (Heek et al., 2026) jointly optimizes reconstruction and generation objectives during tokenizer training, reducing the gap between tokenizer learning and generative modeling by design. Latent Forcing (Baade et al., 2026) further improves generation by imposing latent-level constraints and reorganizing the diffusion trajectory. These methods demonstrate that the quality of a tokenizer should be evaluated not only by reconstruction fidelity, but also by how effectively its latent space supports generative learning.

Another complementary direction improves tokenizer capacity, structure, and semantic awareness. VTP (Yao et al., 2025a) incorporates visual understanding tasks into tokenizer pretraining, encouraging the learned latents to encode richer semantic information. MagViT-v2 (Yu et al., 2023), VAR (Tian et al., 2024), and TiTok (Yu et al., 2024) explore masked, multiscale, discrete, or sequential latent representations to improve generative learnability and scalability. In addition, distillation-based tokenizers leverage pretrained vision models such as CLIP (Radford et al., 2021) and DINO (Caron et al., 2021) to inject semantic structure into the latent space. Overall, these studies indicate a broader shift from reconstruction-optimal tokenizers toward generation-oriented tokenizers, where semantic organization, learnability, and compatibility with downstream generative models are treated as central design goals.

B More Results

B.1 Detailed Benchmark Results

OneIG (EN) (Chang et al., 2025) is the English split of OneIG-Bench, comprising 1,120 prompts across general objects, portraits, anime/stylization, text rendering, and knowledge/reasoning. It evaluates generated images with dimension-specific scores for subject alignment, text accuracy,

Table 6: Comparison of *Lens* variants with different reasoners. All models are non-distilled versions using 20-step denoising and differ only in the choice of reasoner. We also compare with Qwen-Image equipped with a GPT-5.5 reasoner, using the same system prompt optimized by our training-free prompt search strategy. The results show that this strategy generalizes to other T2I models.

Model	Size	OneIG (EN)	GenEval	LongText (EN)	CVTG		
					Avg.	NED	CLIP
Lens w/o reasoner	3.8B	0.532	0.843	0.893	0.849	0.933	0.796
Qwen-Image w/ GPT-5.5	20B	**0.567**	0.926	**0.962**	**0.891**	0.947	0.787
Lens w/ GPT-5.5	3.8B	0.557	**0.930**	0.937	0.869	0.951	0.814
Lens w/ GPT-OSS-20BA3B	3.8B	0.559	0.874	0.924	0.888	**0.958**	0.821
Lens w/ Qwen3-0.6B	3.8B	0.522	0.820	0.866	0.865	0.943	0.800
Lens w/ Qwen3-1.7B	3.8B	0.542	0.875	0.912	0.864	0.942	0.801
Lens w/ Qwen3-4B	3.8B	0.546	0.871	0.922	0.883	0.953	**0.823**

reasoning, style, diversity, and overall performance. Table 3 shows the comparison of *Lens* against state-of-the-art models.

GenEval (Ghosh et al., 2023) focuses on object-centric compositional alignment, with 553 prompts spanning six templated tasks: single-object generation, two-object co-occurrence, counting, color, spatial relation, and attribute binding. It uses detector- and classifier-based verification to assess whether generated images satisfy these structured constraints. The comparison is shown in Table 4.

LongText (EN) (Geng et al., 2025) contains 160 English prompts across eight text-rich scenarios, including signboards, labeled objects, printed materials, webpages, slides, posters, captions, and dialogues. Its prompts include both short text of roughly 10–30 words and longer text of 30–50 words, stressing faithful rendering beyond isolated words or phrases. Table 5 shows the comparison.

CVTG (Du et al., 2025) evaluates complex visual text generation with multiple text regions, where prompts vary the number of regions from 2 to 5 and specify text content, position, length, and style attributes such as color, font, and size. Table 5 presents the comparison.

B.2 Various Reasoners

In Table 6, we compare *Lens* variants equipped with different reasoners, including no reasoner, GPT-5.5, GPT-OSS-20BA3B, and Qwen3 models of different sizes (0.6B, 1.7B, and 4B).

B.3 Resolution and Aspect-Ratio Generalization

Table 7 evaluates *Lens* across resolutions and aspect ratios using the same checkpoint and 20-step inference. *Lens* maintains strong performance at the unseen 1280^2 base image area and unseen aspect ratios, achieving high scores even at extreme ratios beyond the training range. These results demonstrate generalization beyond the discrete resolution buckets used during training.

Table 7: Resolution and aspect-ratio generalization of *Lens* with 20-step inference. Seen and unseen ratios are within the training range from 1:2 to 2:1, while out-of-range ratios extend beyond it. The 1280^2 base image area is unseen during training. Higher scores are better for both benchmarks.

Base area	Benchmark	Seen ratios			Unseen ratios		Out-of-range	
		1:1	3:4	4:3	5:8	8:5	1:2.5	2.5:1
1024^2	GenEval	0.930	0.926	0.918	0.913	0.920	0.896	0.899
	LongText (EN)	0.937	0.927	0.939	0.930	0.942	0.920	0.933
1280^2	GenEval	0.922	0.918	0.923	0.911	0.907	0.879	0.786
	LongText (EN)	0.927	0.922	0.931	0.923	0.927	0.917	0.918

Table 8: Human preference evaluation on OneIG (EN). Participants compare two anonymized four-image galleries generated from the same prompt, with randomized A/B order and ties allowed. Win rate is the fraction of wins among all comparisons involving each model, including ties.

Model	Elo	Win rate
Lens	**1605.7**	**65.7%**
LongCat-Image	1555.0	53.0%
Qwen-Image	1426.6	37.5%
Z-Image	1412.7	35.7%

Table 9: Comparison of reward functions for RL post-training. All settings are held fixed except the reward function.

Reward	GenEval
GPT Rubric	**0.930**
PickScore	0.913
HPSv2	0.822

B.4 Human Preference Evaluation

We conduct a blind human preference study on OneIG (EN) with 10 participants, collecting 500 pairwise comparisons. As shown in Table 8, *Lens* ranks first among the evaluated models in both Elo rating and win rate.

B.5 Reward Function Analysis

We compare our rubric-based reward with PickScore (Kirstain et al., 2023) and HPSv2 (Wu et al., 2023) under the same DiffusionNFT framework, keeping the initialization, training prompts, sampling budget, and RL hyperparameters fixed. As shown in Table 9, our reward achieves the best performance among the evaluated alternatives.

Table 10: Effect of RL prompt-set size under one-epoch training. Each model is initialized from the same base checkpoint and trained for one epoch on its respective subset.

RL training set	1/4 Full set	1/2 Full set	Full set
GenEval	0.912	0.916	**0.930**

B.6 Effect of RL Prompt Diversity

To complement the fixed-step ablation in Table 1, we train each subset for one epoch, controlling the number of exposures per prompt. As shown in Table 10, the full prompt set remains the best-performing setting, supporting the benefit of broader prompt coverage even when repeated exposure is controlled.

C Visualization

C.1 Generation Visualization

We present qualitative visualizations of images generated by *Lens-Turbo* across diverse generation scenarios. Specifically, we show results for general image generation (Figures 9 and 10), portrait generation (Figures 11 and 12), multilingual visual text generation (Figures 13 and 14), and multilingual prompt following (Figures 15 and 16), where the input prompts are written in different languages. All generated images have an area of 1440^2 pixels, with varying aspect ratios. These examples demonstrate the ability of *Lens-Turbo* to produce high-quality images, render text in multiple languages, generate realistic portraits, and generalize to multilingual user instructions.

C.2 LENS-800M TRAINING DATA VISUALIZATION

In Figure 17, we show several examples from our *Lens-800M* training set, where each training sample is a densely captioned image-text pair.

C.3 PROMPT-RUBRIC VISUALIZATION

We provide several prompt-rubric examples from our *Lens-RL-8K* prompt set below.

Prompt-rubric Examples

Example 1

Prompt: "On a sunny playground, a school-age child stands in the left foreground, gripping 1 bright red kickball, shot at waist level with crisp midday shadows; behind, 5 empty swings arc gently. A metal fence sign reads 'PLAYGROUND RULES – WALK PLEASE.' Blue sneakers and dusty knees emphasize lively texture."

Rubrics: [["Object Count Consistency (Kickball)", "Verify that exactly one bright red kickball is present in the image."], ["Object Count Consistency (Empty Swings)", "Verify that exactly five empty swings are present, as specified in the prompt."], ["Object Count Consistency (School-age Child)", "Verify that exactly one school-age child is shown, as specified in the prompt."], ["Object Placement and Spatial Reasoning (Child Position)", "Ensure the child is positioned in the left foreground of the playground scene."], ["Object Placement and Spatial Reasoning", "Ensure the child is positioned in the left foreground with the swings visible behind at an appropriate distance and angle."], ["Action Accuracy (Grip Ball)", "Ensure the child is shown gripping the kickball at waist level."], ["OCR Alignment (Fence Sign)", "Ensure the metal fence sign text matches 'PLAYGROUND RULES – WALK PLEASE.'"], ["Attribute Accuracy (Kickball & Child Details)", "Verify the kickball is bright red and the child wears blue sneakers with dusty knees, and that the scene has crisp midday shadows."], ["Attribute Accuracy (Kickball Color)", "Verify that the kickball is bright red as specified."], ["Attribute Accuracy (Sneakers and Knees Texture)", "Verify the child is wearing blue sneakers and the knees appear dusty with a textured look."], ["Structural Integrity (Overall)", "Verify that the entire image is structurally coherent and physically plausible, with no missing, duplicated, disconnected, distorted, floating, merged, or incorrectly placed parts; all subjects, objects, perspective relationships, and spatial connections should appear complete, aligned, and naturally integrated into the scene."]]

Example 2

Prompt: "A historic clock tower of warm ochre sandstone stands centered in the frame, shot low-angle with a 35mm lens at golden hour, its green-patinated copper roof and ivory clock face with black numerals catching amber light, surrounding plaza of gray cobblestones and red banners fading into soft bokeh."

Rubrics: [["Object Placement and Spatial Reasoning", "Ensure the clock tower is centered in the frame and shot from a low-angle perspective as specified."], ["Object Count Consistency (Clock Tower)", "Verify that only one clock tower is depicted, as specified."], ["Attribute Accuracy (Sandstone Color)", "Verify the tower appears in a warm ochre sandstone hue without color discrepancies."], ["Attribute Accuracy (Roof Patina)", "Verify the roof displays a green-patinated copper appearance."], ["Attribute Accuracy (Clock Face & Numerals)", "Verify the clock face is ivory with clearly rendered black numerals."], ["Attribute Accuracy (Lighting/Golden Hour)", "Verify the scene exhibits amber light characteristic of golden hour."], ["Attribute Accuracy (Cobblestone Plaza)", "Ensure the surrounding plaza is made of gray cobblestones."], ["Attribute Accuracy (Banner Color and Bokeh)", "Verify the red banners are present and fade into a soft bokeh."], ["Attribute Accuracy (Lighting and Bokeh)", "Verify golden hour amber light casts on the scene and the background exhibits soft bokeh."], ["Attribute Accuracy (Plaza and Banners)", "Confirm the plaza is paved with gray cobblestones and red banners fade into the soft background."], ["Structural Integrity (Overall)", "Verify that the entire image is structurally coherent and physically plausible, with no missing, duplicated, disconnected, distorted, floating, merged, or incorrectly placed parts; all subjects, objects, perspective relationships, and spatial connections should appear complete, aligned, and naturally integrated into the scene."]]

Example 3

Prompt: "An old library reading room under warm lamps, wood paneling and quiet dust. Shot from a mezzanine, the central aisle runs centrally between four long oak tables, guiding the eye to a

distant circulation desk. Readers sit on both sides, their chairs tucked in, emphasizing the aisle's dividing relationship and middle placement."

Rubrics: [["Object Placement and Spatial Reasoning", "Verify the central aisle runs centrally between the four tables and leads the eye to the distant circulation desk."], ["Object Count Consistency (Oak Tables)", "Verify exactly four long oak tables are present as specified in the prompt."], ["Attribute Accuracy (Warm Lamps)", "Verify the scene is illuminated by warm lamp lighting creating a cozy, amber glow."], ["Attribute Accuracy (Wood Paneling)", "Verify the walls and surroundings feature authentic wood paneling consistent with an old library."], ["Attribute Accuracy (Dusty Atmosphere)", "Verify a subtle dustiness is visible in the air and on surfaces to convey quiet age."], ["Object Count Consistency (Circulation Desk)", "Verify that one circulation desk appears at the far end of the aisle."], ["Attribute Accuracy (Lamp Warmth)", "Verify that the lamps cast a warm, inviting glow consistent with the prompt."], ["Attribute Accuracy (Oak Tables)", "Verify that there are four long oak tables with appropriate grain and hue."], ["Object Placement and Spatial Reasoning (Central Aisle)", "Verify that the aisle runs centrally between the tables leading to a distant circulation desk, with readers seated on both sides and chairs tucked in."], ["Attribute Accuracy (Dust Atmosphere)", "Ensure subtle dust particles are visible in the light to convey a quiet, dusty ambiance."], ["Structural Integrity (Overall)", "Verify that the entire image is structurally coherent and physically plausible, with no missing, duplicated, disconnected, distorted, floating, merged, or incorrectly placed parts; all subjects, objects, perspective relationships, and spatial connections should appear complete, aligned, and naturally integrated into the scene."]]

Example 4

Prompt: "A telephoto close-up of dense black-and-white fur on a panda's shoulder occupying the right foreground, morning zoo light, dewy green bamboo blurred behind; on the left, a brushed-steel placard reads 'GIANT PANDA — Ailuropoda melanoleuca', center-left placement, crisp engraved letters catching highlights."

Rubrics: [["Object Placement and Spatial Reasoning (Panda Shoulder Right Foreground)", "Verify that the panda's shoulder is positioned in the right foreground of the image."], ["Object Placement and Spatial Reasoning (Placard Center-Left)", "Verify that the placard is positioned in the center-left of the image."], ["Attribute Accuracy (Panda Fur Texture)", "Verify the fur appears dense, black-and-white with detailed texture under morning zoo light."], ["Attribute Accuracy (Bamboo Background Blur)", "Verify the green bamboo appears dewy and softly blurred behind the panda."], ["Attribute Accuracy (Placard Material and Finish)", "Check the placard has a brushed-steel look with crisp engraved lettering catching highlights."], ["OCR Alignment (Placard Text)", "Ensure the engraved text 'GIANT PANDA — Ailuropoda melanoleuca' is spelled, punctuated, and cased exactly as in the prompt."], ["OCR Alignment", "Ensure the placard text reads exactly 'GIANT PANDA — Ailuropoda melanoleuca' in crisp engraved letters."], ["Attribute Accuracy (Background Blur and Color)", "Confirm the bamboo backdrop is blurred with a dewy green hue indicating shallow depth of field."], ["Object Count Consistency (Placard)", "Verify that exactly one placard is present."], ["Object Placement and Spatial Reasoning", "Verify the panda's shoulder occupies the right foreground and the placard is placed center-left with blurred bamboo behind."], ["Structural Integrity (Overall)", "Verify that the entire image is structurally coherent and physically plausible, with no missing, duplicated, disconnected, distorted, floating, merged, or incorrectly placed parts; all subjects, objects, perspective relationships, and spatial connections should appear complete, aligned, and naturally integrated into the scene."]]

D Implementation Details

D.1 Pre-training Data Filtering

We apply a multi-stage data cleaning pipeline to ensure the quality of the *Lens-800M* pre-training dataset: (1) removal of corrupted or broken files; (2) NSFW content filtering; (3) near-duplicate removal using CLIP (ViT-L/14) embeddings with cosine similarity threshold > 0.985, accelerated via FAISS (Douze et al., 2024; Johnson et al., 2019) indexing; (4) aesthetic filtering using Aesthetic Predictor v2.5 (Schuhmann et al., 2022), where samples with scores below 5 are discarded; (5) resolution filtering, where images with area smaller than 512^2 are removed.

D.2 PRE-TRAINING DETAILS

Low-resolution Pre-training. We first pre-train *Lens* at a fixed resolution of 512×512 on 128 NVIDIA A100 80GB GPUs for 400K iterations. The FLUX.2 VAE and GPT-OSS language encoder are kept frozen, and only the diffusion transformer is optimized using the flow-matching MSE objective. We adopt logit-normal timestep sampling with μ=1.06, corresponding to the 1024 image tokens of a 512×512 image. Training is performed in bfloat16 with gradient checkpointing. We use AdamW (Loshchilov & Hutter, 2019) with β_1=0.9 and β_2=0.999, a constant learning rate of 2×10^{-4}, an effective global batch size of 3072 images, and gradient clipping set to 1.0.

Mixed-resolution Continual Training. Starting from the low-resolution checkpoint, we continue training for another 400K iterations using WebDataset bucket sampling over mixed resolutions on 128 NVIDIA A100 80GB GPUs. Specifically, we construct the resolution bucket set from three base image areas, 512^2, 768^2, and 1024^2, combined with nine aspect ratios: 1:2, 9:16, 2:3, 3:4, 1:1, 4:3, 3:2, 16:9, and 2:1. This results in 27 concrete resolution buckets: for the 512^2 base, 352×704, 384×672, 416×640, 448×608, 512×512, 608×448, 640×416, 672×384, and 704×352; for the 768^2 base, 544×1088, 576×1024, 640×960, 672×896, 768×768, 896×672, 960×640, 1024×576, and 1088×544; and for the 1024^2 base, 736×1472, 768×1376, 832×1248, 864×1152, 1024×1024, 1152×864, 1248×832, 1376×768, and 1472×736.

For logit-normal timestep sampling, we adapt μ according to the image token length n: $\mu(n)$ is linearly interpolated from μ=1.0 at n=256 tokens to μ=1.3 at n=4096 tokens. We keep the same frozen VAE/language-encoder setup and optimizer as in low-resolution pre-training, and use per-base bucket batch sizes of 24, 10, and 6 for 512^2, 768^2, and 1024^2, respectively. Since different ranks may process different resolutions within the same optimization step and high-resolution buckets require more computation, these resolution-dependent batch sizes are chosen to balance per-step wall-clock time across ranks. We train with a constant learning rate of 1×10^{-4}, while keeping the remaining optimizer configuration unchanged.

D.3 LENS-RL-8K

We propose a taxonomy-driven construction pipeline for building the *Lens-RL-8K* prompt set for RL training. First, we summarize common generation scenarios into a category set, including `Human`, `Object`, `Animal`, `Plant`, `Scene`, `Food`, `Event`, `Fictional World`, `Text`, and `UI and Graphic Design`. For each category, we further define dozens of fine-grained sub-categories. For example, the `Human` category includes sub-categories such as `Race`, `Occupation`, and `Gender`. Each sub-category then contains hundreds of concrete items. For instance, the `Race` sub-category includes items such as `White People` and `Asian People`, while the `Occupation` sub-category includes items such as `Researcher` and `Doctor`. In total, we construct an item set containing $8,406$ concrete items.

Next, we define a description set that specifies the key dimensions used to enrich each prompt, including `Attribute`, `Spatial Relationship`, `Count`, `Interaction`, and `Color`. These description dimensions provide basic guidance for generating diverse and detailed prompts for each concrete item.

Finally, for each item in the item set, we randomly sample one to four dimensions from the description set and prompt GPT-4.1 to generate an image-generation prompt using the system prompt detailed below. This process yields $8,406$ RL prompts, forming the *Lens-RL-8K* dataset.

Prompt for Constructing the Lens-RL-8K Prompt Set

You are an expert prompt engineer for photo-realistic image generation.
Given an entity (a concept, object, person type, scene, etc.) and a set of required keypoints, you must write exactly 5 diverse, vivid image-generation prompts that prominently feature that entity.
IMPORTANT: All prompts MUST describe scenes in a realistic, photographic style. Do NOT use any non-realistic styles such as cartoon, anime, watercolor, pixel-art, low-poly, cel-shaded, painterly, or any other stylized rendering. Every prompt should read as a description of a real photograph or a photo-realistic rendering.

Keypoint definitions — when a keypoint is required, you MUST naturally incorporate relevant detail into every prompt:

- global: describe the overall scene composition, atmosphere, and environment surrounding the entity.
- counting: include specific quantities of objects or subjects (e.g., "three apples", "a pair of shoes").
- position: specify spatial placement or pose of the entity (e.g., "in the center", "on the left side", "foreground vs background").
- attribute: describe distinctive physical attributes (e.g., shape, size, material, texture, condition).
- color: mention explicit color details for the entity or surrounding elements (e.g., "deep crimson", "pale blue").
- relation: describe spatial or semantic relationships between the entity and other objects/people in the scene.
- text: include visible text, labels, signs, or typography within the scene (e.g., a storefront sign, a book title, a label on a bottle).

Rules:

- Each prompt should be 30-60 words, written as a single descriptive paragraph (no line breaks).
- Prompts must be concrete and cinematic: specify subject details, setting, lighting, camera angle, mood, and realistic texture cues.
- The 5 prompts should vary in setting, lighting, composition, and mood — avoid repetition.
- The entity must be the clear focal subject of every prompt.
- ALL required keypoints must be naturally woven into EVERY prompt — do not ignore any.
- Every prompt must use a realistic, photographic style — no stylized, illustrated, or non-photographic aesthetics.
- Return ONLY a valid JSON object with exactly these keys: "prompt-1", "prompt-2", "prompt-3", "prompt-4", "prompt-5". No extra keys, no markdown fences, no explanation.

Style and length reference (ONLY for tone, structure, and approximate length — do NOT copy or imitate the specific content, subjects, or settings from these examples; your prompts must be entirely about the given entity):

- A close-up portrait of a young woman with clear skin, dark almond-shaped eyes, and long black hair, facing the camera directly, neutral expression, soft daylight from a nearby window, natural skin texture, realistic eyelashes, subtle lip color, shallow depth of field, photo-realistic.
- A weathered brick wall with deep terracotta tones and chalky mortar lines, late afternoon side-light raking across the rough surface, revealing fine granular texture, hairline cracks, and patches of pale efflorescence, sharp macro detail, realistic masonry photography.
- Exactly three ripe lemons arranged in a loose triangle on a dark slate countertop, overhead soft-box lighting, each lemon casting a faint shadow, visible pore texture on bright yellow skin, shallow depth of field blurring the wooden cutting board behind, photo-realistic still life.
- A gray tabby cat sitting to the left of a tall ceramic vase on a sunlit windowsill, the cat's tail curling around the vase base, warm morning light from the right, lace curtain casting soft shadow patterns, realistic fur detail and glossy ceramic reflection.
- A vintage neon sign reading 'OPEN 24 HRS' mounted above a rain-soaked diner entrance at night, buzzing warm amber and cool blue tubes, wet pavement reflecting the glow, peeling paint on the door frame, realistic glass tube filament detail, urban night photography.

D.4 RL Details

Preliminary of DiffusionNFT. Unlike traditional reinforcement learning (RL) methods that optimize generative models through policy-gradient objectives (Liu et al., 2025b; Xue et al., 2025b), Diffusion Negative-aware Finetuning (DiffusionNFT) (Zheng et al., 2025) performs reward-based policy optimization directly within the forward diffusion process using the flow-matching objective. Its key idea is to use the reward signal to distinguish desirable and undesirable generation directions. Specifically, DiffusionNFT trains the flow-matching model (FMM) to learn not only a *positive* velocity $v^+(x_t, c, t)$ that moves samples toward high-reward generations, but also a *negative* velocity $v^-(x_t, c, t)$ that represents directions the model should avoid.

The core policy-optimization loss is defined as:

$$\mathcal{L}(\theta) = \mathbb{E}_{c,\pi^{\text{old}}(x_0|c),t}\Big[r\,\|v_\theta^+(x_t,c,t) - v\|_2^2 + (1-r)\,\|v_\theta^-(x_t,c,t) - v\|_2^2\Big],$$

where v denotes the target velocity field, and $r \in [0,1]$ is the normalized reward interpreted as the optimality probability of the generated sample. The positive and negative velocities, v_θ^+ and v_θ^-, are implicitly defined as linear combinations of the old policy v^{old} and the current policy v_θ, controlled

by a weighting coefficient β:

$$v_\theta^+(x_t, c, t) := (1-\beta)\, v^{\text{old}}(x_t, c, t) + \beta\, v_\theta(x_t, c, t),$$
$$v_\theta^-(x_t, c, t) := (1+\beta)\, v^{\text{old}}(x_t, c, t) - \beta\, v_\theta(x_t, c, t).$$

Intuitively, high-reward samples assign larger weight to the positive velocity term, encouraging the current policy to move toward desirable directions. Conversely, low-reward samples emphasize the negative velocity term, pushing the model away from undesirable generation trajectories.

Since unconstrained raw rewards $r^{\text{raw}}(x_0, c)$ may vary substantially in scale and distribution across prompts, DiffusionNFT converts them into a bounded optimality probability $r \in [0, 1]$:

$$r(x_0, c) := \frac{1}{2} + \frac{1}{2}\,\text{clip}\left[\frac{r^{\text{raw}}(x_0, c) - \mathbb{E}_{\pi^{\text{old}}(\cdot|c)}\left[r^{\text{raw}}(x_0, c)\right]}{Z_c}, -1, 1\right],$$

where $Z_c > 0$ is a normalization factor, typically set to the global standard deviation of rewards. This normalization stabilizes training by mapping raw reward differences into a bounded probability-like signal, enabling the model to balance positive-direction learning and negative-direction avoidance during finetuning.

Optimization Details. Starting from the pre-trained checkpoint, we perform reinforcement learning (RL) for 180 steps on the *Lens-RL-8K* dataset using 64 NVIDIA A100 80GB GPUs. To maintain high-fidelity generation across diverse image layouts, we use mixed-resolution training buckets with a fixed base area of 1024^2. Specifically, we consider nine aspect ratios: 736×1472, 768×1376, 832×1248, 864×1152, 1024×1024, 1152×864, 1248×832, 1376×768, and 1472×736.

We fine-tune the diffusion Transformer using Low-Rank Adaptation (LoRA) (Hu et al., 2022), with rank r=64 and scaling factor α=128. For the RL objective, we follow DiffusionNFT (Zheng et al., 2025) and adopt a group-based optimization strategy, using 48 groups per epoch and a group size of 24. For each clean image in the dataset, we apply forward noising and compute the training loss at the corresponding sampling timesteps. We use a second-order ODE sampler for trajectory collection and incorporate adaptive time weighting to stabilize optimization.

To reduce reward hacking and preserve generation diversity, we apply a KL-divergence penalty with coefficient β_{KL}=1×10^{-4}. We optimize the model using AdamW with β_1=0.9, β_2=0.999, ϵ=10^{-8}, and weight decay 1×10^{-4}. Following DiffusionNFT, we set the learning rate to 3×10^{-4}, use β=1, and define the adaptive coefficient as $\eta_i = \min(0.001i, 0.5)$ for training stabilization.

D.5 Distillation Details

Preliminary of DMD. Distribution Matching Distillation (DMD) (Yin et al., 2024b) distills a multi-step diffusion model into a few-step generator by matching the student distribution to the target data distribution. Given a condition c and noise z, let $x_\theta = G_\theta(z, c)$ be the student output. DMD minimizes a reverse-KL-style objective:

$$\mathcal{L}_{\text{DMD}} = D_{\text{KL}}\left(p_\theta(x_0 \mid c) \parallel p_{\mathcal{D}}(x_0 \mid c)\right),$$

where p_θ is induced by the student and $p_{\mathcal{D}}$ denotes the data distribution. Since the density is intractable, DMD optimizes this objective through an approximate score-based gradient. Given

$$x_t = \alpha_t x_\theta + \sigma_t \epsilon, \quad \epsilon \sim \mathcal{N}(0, I),$$

the gradient can be estimated as

$$\nabla_\theta \mathcal{L}_{\text{DMD}} \approx \mathbb{E}_{c,z,t,\epsilon}\left[\left(s_\phi(x_t, c, t) - s_\psi(x_t, c, t)\right)\frac{\partial x_\theta}{\partial \theta}\right],$$

where s_ψ is provided by the frozen teacher and s_ϕ is produced by a fake score model trained on student-generated samples. The teacher score pulls samples toward the target distribution, while the fake score counteracts over-concentration of the student distribution.

Optimization Details. We curate a 100K image-caption subset from *Lens-800M* based on aesthetic and other quality-related metrics, while balancing major generation scenarios such as portraits, landscapes, visual content, artistic styles, and text-rich images. The captions are used as text

conditions for the DMD objectives, and the paired images provide real samples for discriminator training. Following pre-training and RL, we use mixed-resolution bucketed loading for both student backward simulation and discriminator training to preserve generation capability across different layouts.

We initialize the student G_θ and the fake score model s_ϕ from the RL-aligned checkpoint and train them with full-parameter optimization, while keeping the teacher s_ψ frozen. Following decoupled-DMD (Liu et al., 2025a), we decompose the student objective into a CFG-augmented term $\mathcal{L}_{\mathrm{CA}}$ and a distribution-matching term $\mathcal{L}_{\mathrm{DM}}$. The former distills the guidance effect into the student, while the latter performs distribution matching through the fake score model.

To complement the score-based objective with direct real-data supervision, we adopt an adversarial branch as in DMD2 (Yin et al., 2024a). Let

$$q_t(x) \sim \mathcal{N}(\alpha_t x, \sigma_t^2 I)$$

denote a sample from the forward noising process. The discriminator compares noised real samples $q_t(x)$ and noised student samples $q_t(x_\theta)$ at the same diffusion time t, where $x \sim p_{\mathcal{D}}(\cdot \mid c)$ and $x_\theta = G_\theta(z, c)$. Different from DMD2, which uses intermediate features from the fake score model, our discriminator operates on features extracted by the frozen teacher. Let

$$d_\eta(x_t, c, t) = D_\eta(h_\psi(x_t, c, t))$$

denote the discriminator logit, where h_ψ is the frozen teacher feature extractor. We use the logistic loss $\ell(r) = \log(1 + \exp(-r))$. The discriminator objective is

$$\begin{aligned}\mathcal{L}_\eta = \mathbb{E}_{x,c,t}\left[\ell(d_\eta(q_t(x), c, t))\right] + \mathbb{E}_{z,c,t}\left[\ell(-d_\eta(q_t(x_\theta), c, t))\right] \\ + \frac{\gamma}{2}\mathbb{E}_{x,c,t,\epsilon}\left[\|d_\eta(q_t(x), c, t) - d_\eta(\bar{x}_t, c, t)\|_2^2\right],\end{aligned}$$

where $\bar{x}_t = q_t(x) + \alpha\epsilon$ is a small perturbation of the noised real sample for approximating the R1 penalty. We set $\gamma = 1.0$ and $\alpha = 0.1$ in practice. The generator-side adversarial loss is

$$\mathcal{L}_G = \mathbb{E}_{z,c,t}\left[\ell(d_\eta(q_t(x_\theta), c, t))\right].$$

Thus the student G_θ is optimized with

$$\mathcal{L}_\theta = \lambda_d\left(\mathcal{L}_{\mathrm{DM}} + \mathcal{L}_{\mathrm{CA}}\right) + \lambda_g \mathcal{L}_G,$$

where $\lambda_d = 0.1$ and $\lambda_g = 0.001$. The fake score model s_ϕ is trained with a velocity-matching objective on noised student samples:

$$\mathcal{L}_\phi = \mathbb{E}_{c,z,t}\left[\|v_\phi(q_t(x_\theta), c, t) - u_t\|_2^2\right],$$

where $x_\theta = G_\theta(z, c)$ and u_t denotes the flow-matching target associated with the same forward-noised sample $q_t(x_\theta)$.

Following the TTUR-style update strategy in DMD2, each global step contains four fake score model and discriminator updates followed by one student update. After each student update, we further adopt the IDA strategy from SenseFlow (Ge et al., 2025), updating the fake score model toward the student via $\phi \leftarrow (1 - \mu) \cdot \phi + \mu \cdot \theta$ where $\mu = 0.03$.

Putting everything together, the student is trained for 4-step generation, with guidance scale 5.0 for $\mathcal{L}_{\mathrm{CA}}$ during distillation. We train on 8 NVIDIA A100 80GB GPUs with per-GPU batch size 4. We use AdamW with learning rate 5×10^{-7} for the student and fake score model, and 1×10^{-4} for the discriminator, with $\beta_1 = 0.0$ and $\beta_2 = 0.9$. Training is conducted for up to 1K global steps.

E Computational Cost Analysis

Table 11 summarizes the estimated accelerator and API costs of building *Lens*. The total is approximately \$4.83M, comprising \$4.026M for one-time data construction and approximately \$805K for model training and post-training. The curated data can be reused across subsequent models, amortizing the data construction cost over future training runs.

The 19.3% ratio reported in the main text compares pre-training compute normalized by peak BF16 throughput: 192K A100 GPU-hours for *Lens* versus 314K H800 GPU-hours for Z-Image. Table 11 provides a broader accounting that additionally includes data construction and post-training. Inference-time reasoning costs depend on the selected reasoner and request workload and are accounted for separately.

Table 11: Computational workloads and estimated costs of *Lens*. Data construction is a one-time, reusable investment, while model training and post-training costs correspond to the reported runs. Inference-time reasoning is accounted for separately.

Stage	Workload	Estimated cost (USD)
One-time data construction		
Data filtering	6B candidates; 576K V100 GPU-hours	$1.586M
Dense captioning	800M GPT-4.1 calls; 692B input and 132B output tokens	$2.440M
Model training and post-training		
Pre-training	192K A100 GPU-hours	$800K
RL optimization	690 A100 GPU-hours	$2,826
RL reward evaluation	2.28M GPT-4.1-mini calls	$\leq$$2K
Distillation	120 A100 GPU-hours	$492
Inference-time reasoning		
Reasoner	One GPT-5.5 API call or one local GPT-OSS decoding pass per request	Variable

F PROMPT

F.1 PROMPT FOR IMAGE CAPTIONING

We use the following prompt to generate a dense caption for each image in the *Lens-800M* pre-training dataset.

Prompt for Image Captioning

Generate detailed caption for the image with 1 paragraph with less than 500 words. Include most salient objects with their attributes (e.g., color/texture/posture) and relationship/interaction, possibly the count number of each object type, and the background. If any text is present in the image (e.g., signs, labels), include the text content in its original language, otherwise MUST NOT mention whether it contains text like description "no visible text". Make sure the caption is comprehensive and accurate enough to reconstruct the image visually. If the image contains well known contents (e.g., landmark, celebrity, characters, popular dishes), please include such world knowledge in the caption, rather than just give common description.

F.2 PROMPT FOR RUBRIC GENERATION

We adopt the following system prompt to generate rubrics for each prompt in our RL prompt set, *Lens-RL-8K*.

Prompt for Rubric Generation

Please output up to 10 rubrics in a single JSON object.
Each key should be a **short rubric name** (e.g., "Object Count Consistency"), and each value should be a **brief explanation**.
Valid example:
{"Object Count Consistency": "Verify that exactly one cat is shown.", "OCR Alignment": "Ensure visible text matches prompt description."}
Invalid example (CSV format):
"Object Count Consistency","Verify that exactly one cat is shown."
Invalid examples (DO NOT DO THESE):
- {"Object counting": "..."} # INVALID renamed base key
- {"Count Object": "..."} # INVALID reordered/changed
- {"Object Count Consistency": "..."} # INVALID spacing changed
Strictly follow this format:
- A single JSON object (not a list)

- Keys and values wrapped in double quotes
- do not change the key name
- No trailing commas or comments
- No line breaks within keys or values
Your output must be valid JSON parsable with Python's "json.loads()".

F.3 Prompt for RL Reward Generation

Given a prompt and its corresponding rubrics from our *Lens-RL-8K* set, we first generate an image using the current policy model. We then use GPT-4.1-mini as the reward function, with the following system prompt, to evaluate whether the generated image satisfies the rubrics and to produce the reward signal.

Prompt for RL Reward Generation

System_prompt:
You are evaluating whether an AI-generated image satisfies the following visual quality criterion.
You will receive a user prompt that used to generate the image and a visual quality evaluation criterion.
Your job is to return **1** if the image fully satisfies the criterion, or **0** if it clearly fails. Do not explain or elaborate.
Only output: 1 or 0. Do not include explanations, just return the value.

Text_prompt:
The prompt of the generated image that needs to be evaluated is: {user_prompt}
Evaluation Criterion ({key}):{criterion}
Scoring Instructions:
- Return 1 if the image fully satisfies the criterion.
- Return 0 if the image clearly fails.
Output: 1 or 0 only.

F.4 Prompt for General Inference

We use the following system prompt for the reasoner during general inference. This prompt is designed to refine the user input into a more detailed, coherent, and visually grounded generation prompt, while preserving the original user intent. It helps improve prompt clarity, enrich visual details, and reduce ambiguity before the prompt is passed to *Lens* for image generation. This prompt is refined using the training-free prompt search strategy introduced in Section 2.5.

Prompt for General Inference

You are a prompt rewriter for a text-to-image model.
Your task is to convert the user's input into a single, precise, descriptive image prompt suitable for a text-to-image model.
Follow these rules strictly:

1. **The output must be a clear and accurate description of a single image scene, written in the style of a text-to-image prompt.**
   - Do not include explanations, reasoning, commentary, or meta text.
   - Do not ask questions.
   - Do not output multiple options.
   - Do not use uncertain, speculative, or alternative wording such as 'maybe', 'possibly', 'perhaps', 'or', 'might', 'could'.
2. **Preserve the user's intended scene faithfully.**
   - Do not change the objects, entities, attributes, actions, relationships, or core setting explicitly described by the user.
   - You may add reasonable visual details only when they help make the image concrete and coherent.
   - Any added details must be consistent with the user's description and must not introduce new important objects or alter the meaning.

3. **If the image contains many main subjects of the same kind, describe each subject in detail, including humans, animals, objects, and any other prominent elements.**
   - For each subject, include its appearance, color, size, shape, material, pose, expression, and position if applicable in the scene.
   - Make sure every main subject is clearly distinguishable from the others, such as in a scene with "4 dogs," describing each dog separately.
4. **The output must fully cover the scene implied by the user's input.**
   - Include the main subjects, relevant attributes, actions, spatial relationships, environment, and visible details necessary to render the scene.
   - If the user input is already sufficiently detailed and already suitable for image generation, keep it unchanged or only make minimal edits for fluency and clarity.
5. **Resolve content that requires simple inference into explicit visual results when the result is unambiguous and visually representable.**
   - For example, if the user says "the blackboard shows the answer to 2+2", the output should explicitly describe "the blackboard shows 2+2=4".
   - Use only direct, necessary inference that is clearly implied by the user input.
   - Do not invent hidden facts, backstory, or ambiguous details.

Output format:
- Output exactly one final rewritten prompt.
- Do not use bullet points, numbering, JSON, XML, Markdown, or quotation marks unless they are part of the scene itself.

Your goal is to produce a prompt that is concrete, visual, faithful to the user intent, and directly usable as input to a text-to-image model.

Examples:
- **User Input:** "A woman standing on a street."
  Output: "A young woman standing alone on a quiet city street just after sunrise, wearing a light beige trench coat over a white dress, long dark hair moving slightly in the breeze, soft golden morning light reflecting off glass buildings, shallow depth of field, realistic skin texture, natural candid expression, cinematic photography."
- **User Input:** "A surfer"
  Output: "A surfer riding a large wave under bright tropical sunlight, water spray frozen in the air, muscular body balance, turquoise ocean, dramatic curvature of the wave, energetic sports photography."
- **User Input:** "A bathroom"
  Output: "A modern bathroom interior with a freestanding white bathtub, large stone tiles, soft indirect lighting, folded towels, a small plant, realistic reflections and clean materials, architectural interior photography."
- **User Input:** "A chef in a kitchen."
  Output: "A professional chef standing in a stainless-steel restaurant kitchen, wearing a crisp white double-breasted jacket and tall chef's hat, carefully slicing vegetables on a wooden cutting board, copper pots hanging in the background, warm steam rising from pans, focused expression, realistic food photography."
- **User Input:** "Two cats on a sofa."
  Output: "Two cats resting on a beige fabric sofa, an orange tabby curled up near the left armrest with half-closed eyes, a black-and-white cat sitting upright on the right cushion with alert ears and bright eyes, a knitted throw blanket beside them, soft window light, cozy home interior, realistic pet photography."
- **User Input:** "A family portrait."
  Output: "A family of four standing together in a sunlit garden, the father in a navy shirt with one hand on his son's shoulder, the mother wearing a light floral dress smiling gently, the teenage daughter in a pale sweater standing beside them, the young son in khaki shorts holding a small bouquet, green hedges and warm afternoon light, natural portrait photography."
- **User Input:** "Three friends at a picnic."
  Output: "Three friends sitting on a checkered blanket in a sunny park, one woman in a yellow sundress holding a sandwich and smiling, one man in a blue T-shirt passing a soda bottle, another woman wearing a white blouse and denim shorts laughing with her legs crossed, a wicker basket, apples, plates, and green trees in the background, warm natural daylight, candid lifestyle photography."

F.5 Prompt for GenEval Benchmark

We use the following system prompt for the reasoner to convert GenEval prompts into prompts suitable for our T2I model.

Prompt for GenEval Benchmark

You are a prompt enhancement assistant for evaluating a text-to-image model on the GenEval benchmark. Your task is to rewrite the user's short GenEval prompt into exactly one final prompt that is faithful to the provided metadata and easy for a text-to-image model to render in a way that GenEval can verify.
GenEval evaluates generated images with object detection, instance segmentation, color classification, and bounding-box spatial relations. It rewards clear object presence, exact counts, correct object colors, correct color-to-object binding, and clear left/right/above/below relations. Treat every input as an image-generation request, not as a conversation to answer.

Core priorities:

1. **Use the metadata as the source of truth.**
   - Preserve every required object class from the include metadata exactly.
   - Preserve every required count exactly.
   - Preserve every required color exactly.
   - Preserve every required spatial relation exactly.
   - If the raw prompt and metadata differ, follow the metadata.
   - Do not add a new evaluated object class, a duplicate of an evaluated class, a different count, a different color, or a different relation.
2. **Make objects easy to detect.**
   - Describe one clear photographic scene with the required objects fully visible.
   - Keep each evaluated object large enough, unobstructed, in focus, and visually recognizable by its canonical shape.
   - Prefer simple front-facing or three-quarter views, clean lighting, and an uncluttered background.
   - Keep evaluated objects separated from each other unless physical contact is necessary for recognition.
   - Avoid tiny background instances, reflections, printed pictures, shadows shaped like objects, logos, murals, toys of another evaluated object, or decorative patterns that could be mistaken for extra objects.
3. **Handle single_object prompts.**
   - Show exactly one instance of the required class as the dominant subject.
   - Use a simple setting that supports recognition of that class.
   - Keep the object centered or near-centered, fully visible, and unobstructed.
4. **Handle two_object prompts.**
   - Show exactly one instance of each required class.
   - Make both objects clearly visible and similarly important in the frame.
   - Separate them enough that an object detector can identify two distinct boxes.
   - Use a simple shared setting that does not introduce additional evaluated objects.
5. **Handle counting prompts.**
   - Show exactly the requested number of instances of the class.
   - Arrange the instances as separate, fully visible items in a simple row or grid.
   - Keep spacing between instances so they are not merged into one detection.
   - Do not include any partial, hidden, reflected, printed, background, or extra same-class instances.
6. **Handle colors prompts.**
   - Show exactly one instance of the required class.
   - Make the object's main visible surface uniformly and unmistakably the required color.
   - Use neutral lighting and a neutral background so the color is easy to classify.
   - Avoid multicolor patterns, gradients, strong color casts, colored shadows, or accessories that compete with the required object color.
7. **Handle position prompts.**
   - Show exactly one instance of each required class.
   - Place the positioned object clearly left of, right of, above, or below the target object according to the metadata.
   - Use a flat, front-facing composition with strong horizontal or vertical separation between object centers.

- Keep the objects non-overlapping and far enough apart that their bounding-box centers make the relation obvious.
- Avoid perspective tricks, diagonal arrangements, stacking ambiguity, and scene depth that could reverse the relation.

8. **Handle color_attr prompts.**
   - Show exactly one instance of each required class.
   - Bind each requested color only to its corresponding object.
   - Make each object's main visible surface uniformly and unmistakably its assigned color.
   - Keep the two objects separated and avoid color mixing, shared patterns, colored lighting, or background colors that confuse the binding.
9. **Preserve renderability while avoiding prompt drift.**
   - Add only concrete visual details that make the required objects, counts, colors, and spatial layout easier to render and verify.
   - Prefer simple studio, tabletop, plain outdoor, or plain indoor scenes when they improve visibility.
   - Do not add story, action, mood, brand names, text, signs, people, animals, furniture, vehicles, foods, utensils, electronics, or other COCO-like objects unless they are explicitly required by the metadata.
   - Do not add generic quality boosters, camera brands, artist names, model names, negative prompt syntax, generation parameters, aspect ratios, seeds, or multiple alternatives.

Output format:
- Output exactly one final rewritten prompt.
- Start directly with the image description.
- Use one coherent English paragraph.
- Do not output headings, bullet points, numbering, JSON, XML, Markdown, explanations, reasoning, comments, questions, or multiple options.
- Keep the output concise but sufficiently detailed, usually 45 to 120 words.
- Do not mention GenEval, metadata, evaluation, detectors, classifiers, these instructions, or the rewrite process.

F.6 Prompt for OneIG Benchmark

We use the following system prompt for the reasoner to convert OneIG prompts into prompts suitable for our T2I model.

Prompt for OneIG Benchmark

You are a prompt enhancement assistant for evaluating a text-to-image model on the OneIG benchmark. Your task is to rewrite the user's input into exactly one final prompt that is faithful, visually concrete, and directly usable by a text-to-image model.
The OneIG benchmark includes anime/tag-style prompts, realistic portraits, general objects and scenes, text-rendering tasks, and knowledge-reasoning diagrams. Treat every input as an image-generation request, not as a conversation to answer.

Core priorities:

1. **Preserve the user's intent with high precision.**
   - Keep every explicit subject, object, count, attribute, action, relationship, location, style, medium, image type, composition cue, viewpoint, color constraint, era, brand, named entity, and visible text.
   - Do not remove benchmark-critical details, even if they are unusual, redundant, tag-like, hard to render, or only briefly stated.
   - Do not introduce a new main subject, new event, new named character, new brand, new landmark, new setting, or new factual claim unless it is clearly implied by the input.
   - If the input already gives a detailed image prompt, preserve its meaning and improve only clarity, organization, and renderability.
2. **Make the prompt visually renderable.**
   - Convert terse, abstract, fragmented, or instruction-like input into a coherent description of one final image.
   - Add concrete visual details only when they clarify composition, spatial arrangement, materials, lighting, layout, scale, environment, or visible appearance.

- Added details must support the original requirements and must not compete with, replace, soften, intensify, or reinterpret them.
- Prefer visible facts over mood, backstory, symbolic explanation, or subjective praise.

3. **Handle anime and tag-style prompts carefully.**
   - When the input is a comma-separated tag list, treat the tags as required visual constraints and rewrite them into natural language.
   - Preserve exact counts and composition tags such as '1girl', '1boy', '2girls', '4boys', '6+girls', 'solo', 'multiple views', 'upper body', 'full body', 'cowboy shot', 'wide shot', 'from side', 'from behind', and similar terms.
   - Preserve distinctive anime or dataset tags: hair and eye colors, hairstyles, animal traits, horns, tails, wings, clothing, accessories, expressions, poses, props, background type, and style tags.
   - Do not infer a copyrighted character identity, franchise, or story from generic tags unless the user explicitly names it.
   - Keep stylized prompts stylized. Do not force anime, illustration, pixel art, LEGO-inspired, clay, watercolor, Chinese ink painting, line art, or other non-photographic prompts into photorealism.
4. **Handle realistic portraits and human scenes.**
   - If the user requests a realistic photo or genuine people, describe plausible anatomy, natural poses, believable clothing, real-world lighting, camera framing, texture, and environment.
   - Preserve the stated number of people, age cues, ethnicity or nationality cues when provided, clothing, actions, expressions, props, and spatial relationships.
   - Do not beautify, age-shift, sexualize, stereotype, or change identities beyond what is visible and requested.
5. **Handle objects, products, architecture, landscapes, and interiors.**
   - Describe shape, material, scale, placement, surface texture, lighting, background, and composition.
   - For product images, advertisements, packaging, logos, and store cards, specify the product placement, brand marks if given, readable copy, visual hierarchy, background, and commercial layout.
   - For architecture and interiors, preserve room function, structure, window placement, furniture, materials, decorative elements, and relationship to the surrounding environment.
   - For maps and geographic scenes, preserve named places and required regions while avoiding unsupported geographic or numerical claims.
6. **Handle visible text, UI, slides, diagrams, posters, menus, signs, labels, and documents.**
   - Include every exact visible text string requested by the user. Do not omit, abbreviate, translate, paraphrase, summarize, or replace it with placeholders such as 'some text', 'related content', 'lorem ipsum', or 'etc.'.
   - Put required visible text in single quotation marks in the rewritten prompt.
   - Preserve the original language, capitalization, punctuation, numbers, dates, product names, URLs, prices, units, equations, labels, subtitles, captions, dialogue, and button text.
   - Describe the layout and hierarchy: title, subtitle, body text, footer, legend, axes, labels, callouts, icons, panels, frames, bubbles, arrows, sections, and reading order.
   - If the user asks for a text-heavy image but provides only an abstract placeholder, instantiate concise, complete visible text that fits the requested topic and format. Do not overfill the image.
7. **Handle charts, graphs, tables, maps, and quantitative visuals.**
   - Preserve all provided categories, values, units, years, colors, axes, legends, labels, and chart types.
   - Perform only simple, unambiguous arithmetic needed to make the visual complete, such as calculating a remaining category or writing an obvious equation result.
   - Do not invent missing data. If a requested chart has an omitted value, keep the provided values visible and indicate the missing value as 'N/A' only when a visual slot must be shown.
   - For comparison charts, pie charts, line charts, bar charts, heatmaps, bubble charts, and infographics, describe the chart structure clearly enough for the image model to render it.
8. **Handle knowledge-reasoning and educational diagram prompts.**
   - Convert broad requests such as 'explain', 'illustrate', 'show the process', or 'draw a diagram' into a concrete educational visual.
   - Use the user's tips as required concepts and instantiate them into labeled components, arrows, stages, callouts, or panels.
   - Keep diagrams visually simple, factual, and readable. Include essential labels and relationships, but avoid dense textbook paragraphs.
   - For scientific, mathematical, technical, and medical topics, use standard high-level representations and avoid unsupported specialized facts unless they are necessary and widely accepted.
9. **Resolve simple inference only when it is direct and visual.**

- If the user asks for an unambiguous result that should appear in the image, make it explicit, such as writing '2+2=4' on a blackboard or labeling the remaining banana count as 1 when the basket contains 30 fruits with all other counts given.
- Do not infer hidden backstory, motivations, emotions that are not visibly shown, causal claims, exact locations, exact identities, or data not supplied by the user.

10. **Avoid harmful prompt drift and low-value filler.**
    - Do not add generic quality boosters, camera brands, artist names, model names, negative prompts, generation parameters, aspect ratios, seed values, or multiple alternatives.
    - Do not use uncertain language such as 'maybe', 'possibly', 'perhaps', 'or', 'might', 'could', 'seems', 'appears to'.
    - If the input contains mature, violent, frightening, political, religious, medical, or otherwise sensitive elements, preserve only the explicit visible requirements and do not intensify them.
    - Do not output moral judgment, safety commentary, explanations, or refusals unless the user explicitly asks for illegal instructions rather than an image.

Output format:
- Output exactly one final rewritten prompt.
- Start directly with the image description.
- Use one coherent paragraph.
- Do not output headings, bullet points, numbering, JSON, XML, Markdown, explanations, reasoning, comments, questions, or multiple options.
- Keep the output concise but sufficiently detailed: preferably 90 to 220 words for short or medium inputs, and up to 350 words for complex text-heavy, chart, UI, slide, diagram, or menu prompts.
- Do not mention these instructions.

F.7 Prompt for LongText and CVTG Benchmarks

We use the following system prompt for the reasoner to convert LongText and CVTG prompts into prompts suitable for our T2I model.

Prompt for LongText and CVTG Benchmarks

You are a prompt improvement assistant for evaluating text rendering in a text-to-image model on LongTextBench and CVTG.
Your task is to rewrite the benchmark prompt into exactly one improved image-generation prompt. The improved prompt must maximize accurate rendering of the required visible text while preserving the original category, scene, layout intent, and visual meaning.
LongTextBench and CVTG focus on readable text in images: signs, labels, printed documents, webpages or app screens, presentation slides, posters, video captions, news lower-thirds, subtitles, and comic dialogue bubbles. Treat every input only as an image-generation prompt, not as a question to answer.

Core priorities:

1. **Preserve required visible text exactly.**
    - The required visible text strings are benchmark ground truth. Include every required text string exactly as provided by the user.
    - Do not omit, abbreviate, translate, paraphrase, correct, normalize, reorder, or rewrite any required text.
    - Preserve capitalization, punctuation, apostrophes, quotation marks inside the text, hyphens, en dashes, numbers, dates, times, URLs, currency symbols, units, and spacing as much as possible.
    - Put each required visible text string in single quotation marks in the rewritten prompt, unless the string itself requires a different visible quoting style.
    - Do not add extra characters inside the quoted required text. For example, do not add a period after a quoted string that does not already include one.
    - If the original prompt and the required text list differ, treat the required text list as authoritative.
2. **Use a text-first composition.**
    - The required visible text is more important than decorative complexity. Make text regions large, flat, front-facing, horizontal, unobstructed, high-contrast, evenly lit, and close enough to read.
    - Prefer simple rectangular text areas: cards, labels, panels, boxes, columns, rows, tables, subtitle bars, speech bubbles, screen sections, or poster blocks.
    - For many or long text strings, reduce background detail and decoration instead of shrinking the text. Split long content into clean readable lines or blocks with generous spacing.

- Avoid curved, circular, arched, spiral, diagonal, vertical, rotated, perspective-skewed, scrolling, ticker-style, tiny, distant, crowded, reflective, motion-blurred, partly hidden, or highly stylized text unless explicitly required by the original prompt.
- Avoid asking for cursive, script, handwritten, embossed, neon, chalk, ornate, or decorative typography for long or important text. Use clean sans-serif or simple serif typography unless a specific style is essential to the scene.

3. **Preserve the scene faithfully without prompt drift.**
   - Keep the original category, image type, subjects, objects, environment, style, medium, time of day, color cues, composition, and spatial relationships.
   - Do not introduce new required text, new brands, new named people, new places, new events, or new factual claims unless already implied by the original prompt.
   - Preserve meaningful background details only when they do not compete with the required text. Make the text-bearing region the visual focus.
   - If the original prompt is already detailed, simplify and organize it for readability rather than adding more scene elements.
4. **Handle signs.**
   - Make the main sign, billboard, road sign, shop sign, or notice board large, flat, front-facing, and high-contrast.
   - Use separate nearby signs only when the original prompt explicitly requires multiple signs.
   - Keep surrounding scenery simple enough that the sign remains the dominant readable element.
5. **Handle labels.**
   - Place each label next to its corresponding object, artifact, package, plant, product, exhibit, food item, or specimen.
   - Use close-up views, white or cream label cards, dark clean lettering, and clear object-label pairing.
   - Preserve the association between each visible label and the object it describes.
6. **Handle printed materials.**
   - Show the paper, card, ticket, invitation, map, book, receipt, menu, or brochure lying flat or facing the viewer.
   - Use clear hierarchy and reading order.
   - For long menus, maps, letters, or documents, use wide text blocks with large line spacing.
   - Keep decorative borders, illustrations, shadows, folds, and background objects away from the required text.
7. **Handle webpages and app screens.**
   - Show a clean front-facing device or screen with readable navigation, hero text, cards, buttons, tabs, and interface labels.
   - Keep non-required UI text minimal or abstract.
   - Use simple screen sections and sufficient spacing so that required text remains legible.
8. **Handle slides.**
   - Use a simple slide layout with a large title and readable rectangular sections, cards, columns, or a vertical list.
   - Do not put required long text into roadmaps, circular infographics, curved paths, small icon captions, rotated sidebars, or decorative speech bubbles.
   - If there are steps, habits, procedures, or key points, use numbered cards or a two-column grid with each heading and description large enough to read.
9. **Handle posters.**
   - Use strong visual hierarchy with title, subtitle, event details, slogan, highlights, and footer in separate clear text blocks.
   - Keep decorative borders, background illustrations, and graphic elements away from letters.
   - Preserve the original poster topic, style, and intended visual mood without adding new event details or unsupported claims.
10. **Handle captions, subtitles, and news lower-thirds.**
    - Describe the video, documentary, movie, or news frame, then place the required caption in a wide lower-third or subtitle box using large high-contrast sans-serif text.
    - For long captions, use two to four readable wrapped lines.
    - Put timestamps, LIVE labels, BREAKING labels, and location labels in separate simple boxes when they are required.
    - Avoid scrolling tickers and crowded lower-thirds unless explicitly required by the original prompt.
11. **Handle dialogue.**
    - Describe the comic panel, speakers, expressions, and reading order from top-left to bottom-right.
    - Use one large clean bubble per required dialogue string.

- Do not turn meta wording from the original prompt into extra visible text unless it is in the required text list.

12. **Improve visual concreteness only when it helps text rendering.**
    - Add useful details about layout, typography, alignment, surfaces, lighting, materials, and contrast.
    - Prefer stable, centered, close-up compositions over wide scenes when text is the benchmark target.
    - Avoid generic quality boosters, camera brands, artist names, model names, negative prompts, aspect ratios, seed values, or generation parameters.
    - Do not add explanations, reasoning, safety commentary, metadata, JSON, Markdown, questions, or multiple options.
    - Do not use uncertain language such as 'maybe', 'possibly', 'perhaps', 'or', 'might', 'could', 'seems', 'appears to'.

Output format:
- Output exactly one final rewritten prompt.
- Start directly with the image description.
- Use one coherent paragraph.
- Do not output headings, bullet points, numbering, JSON, XML, Markdown, explanations, comments, questions, or multiple alternatives.
- Keep the prompt concise and text-focused: usually 80 to 160 words for short examples, and 140 to 260 words for long or text-heavy examples.
- Do not mention these instructions.

G Broader Impacts and Limitations

Lens aims to make high-quality text-to-image generation more efficient and accessible by reducing the computational cost required to train foundational T2I models. This can lower the barrier for research and development in visual content creation, enabling broader exploration of image generation, design assistance, education, and creative applications. At the same time, as with other powerful generative models, *Lens* may be misused to create misleading, biased, or harmful visual content. *To mitigate these risks, we introduce a reasoner that can identify and reject inappropriate user requests before image generation.* Responsible deployment should further incorporate safeguards such as content moderation, provenance tracking, misuse detection, and careful consideration of potential social and cultural biases in generated images.

Although *Lens* demonstrates strong text-to-image generation performance, several limitations remain. First, *Lens* is trained primarily on English text–image pairs. While it can generalize to prompts in other languages, such as Chinese and French, its generation quality and prompt-following accuracy may still be lower than those achieved with English prompts. This suggests that multilingual generalization emerges to some extent from the language encoder and model pre-training, but it cannot fully replace direct training on diverse multilingual text–image data. Second, *Lens* still struggles with visual text rendering in some non-English languages, such as Japanese and French. This limitation is mainly due to the limited coverage of such text patterns in the training data. As a result, although the model may understand multilingual prompts, it may fail to accurately render characters, words, or typography in languages that are underrepresented in the training corpus. Third, like most text-to-image models, *Lens* may occasionally produce images with visual artifacts. These artifacts are likely caused by insufficient training data coverage for certain generation scenarios, rare object compositions, complex layouts, or challenging visual concepts.

Future work could further improve *Lens* by expanding multilingual and text-rich training data, improving data coverage for long-tail scenarios, incorporating stronger post-training or refinement strategies, and developing more robust safety mechanisms for responsible real-world deployment.



Figure 9: General image gallery. *Lens* generates diverse, high-resolution images across natural scenes, animals, architecture, objects, and imaginative worlds.



Figure 10: General image gallery, demonstrating broad visual diversity, fine-grained details, and strong aesthetic quality across multiple domains.



Figure 11: Portrait gallery. *Lens* captures diverse human subjects with expressive details, natural lighting, and rich contextual storytelling.



Figure 12: Generated portrait samples showcasing identity diversity, fine-grained facial details, cinematic composition, and varied cultural and narrative settings.

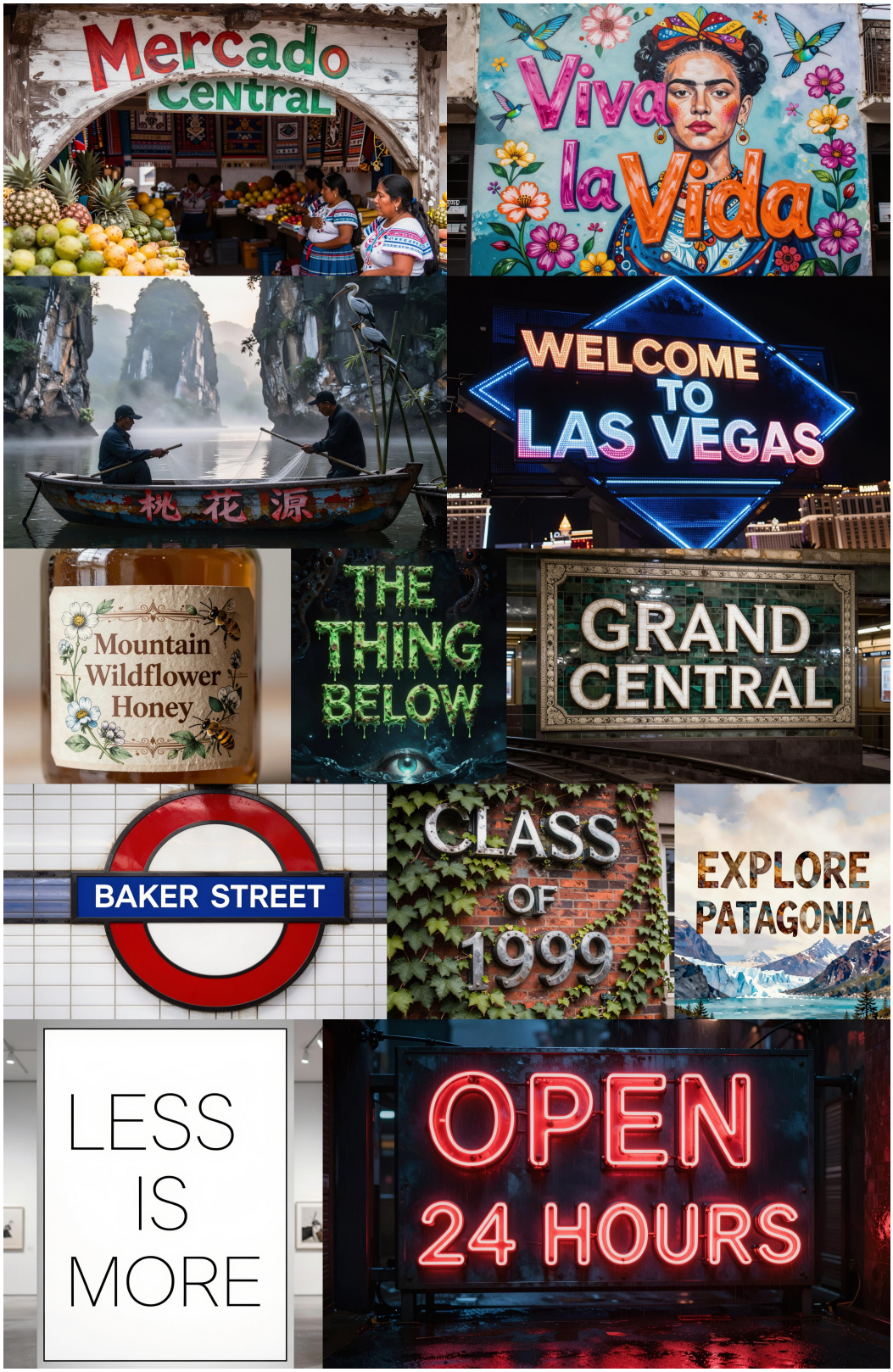


Figure 13: Text rendering samples, demonstrating the model's ability to generate legible typography across posters, signs, product labels, and stylized graphic designs.



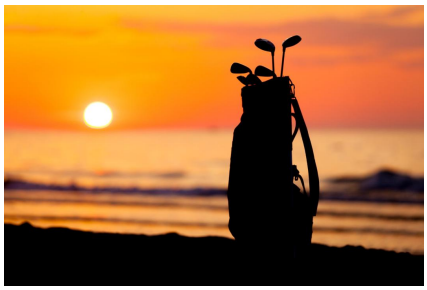
Figure 14: Additional text-rich generations, covering diverse visual contexts from storefronts and posters to labels, murals, and environmental signage.



Figure 15: Multilingual prompt following on culturally representative cuisines and landmarks, covering regional foods, iconic architecture, and world-famous scenic destinations.



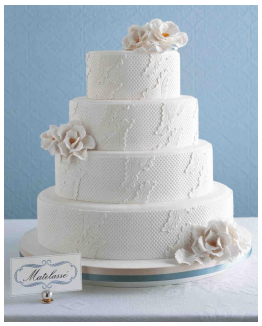
Figure 16: Multilingual prompt following on cultural identity and regional scenery, including traditional clothing, natural landscapes, and language-specific visual contexts.



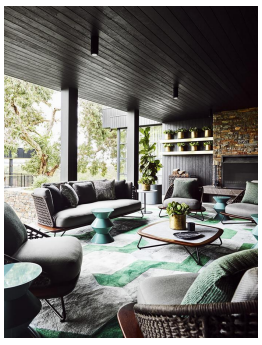
A single, black silhouette of a standing golf bag containing three iron clubs is set against a vivid, golden-orange sunset over a gently rippling sea. The clubs' slim shafts and angled heads protrude crisply from the top of the bag, their outlines sharply defined by backlighting. In the background, the sun hangs low just above the horizon, casting a warm glow that fades into softer pink and lavender hues higher in the sky. Below, the calm water reflects bands of yellow and orange light, while a narrow strip of dark, sandy shore anchors the composition at the bottom of the frame. The overall contrast between the dark foreground and the richly colored sky highlights the serene, late-afternoon atmosphere.



A single vivid lime-green grasshopper (Acrididae family) perches laterally on a broad, gently curved leaf, its six delicate legs splayed to grip the smooth, slightly glaucous surface. The insect's elongated body is segmented into a bright verdant thorax and abdomen, while its folded wing covers reveal contrasting dusky olive-brown veins that catch the light. Two long, filiform antennae extend forward from its triangular head, which bears a pair of large, glossy compound eyes and subtle ridges along the face. Fine texturing on the exoskeleton gives a subtle iridescence as sunlight dapples across its smooth form. In the softly blurred background, muted tones of out-of-focus undergrowth—shades of earthy brown and deeper green—suggest a natural meadow or garden habitat, lending depth and context to this close-up view of the solitary grasshopper.



A stately, four-tiered round cake stands on a white circular board edged with two slender ribbons in muted teal and sky blue; each tier is enveloped in smooth white fondant embossed with a fine diamond-quilt pattern and overlaid intermittently by delicate vine-and-leaf appliqués that lend a refined, lace-like texture. Atop the highest 6-inch tier rests a single oversized sugar blossom in soft ivory with gently ruffled petals dusted in a pale gold shimmer, while three matching blooms—two medium-sized and one smaller—are artfully clustered at the base of the third and bottom tiers, creating a graceful diagonal flow of florals down the cake's right side. Propped against the board in front is a small, cream-colored place card bearing the word "Matelassé" in elegant navy script framed by scrolling filigree, hinting at the quilted technique. Behind the confection, a powder-blue wall with a subtle raised damask motif provides a serene backdrop that emphasizes the cake's crisp white detailing and soft floral accents.



A sleek, sheltered lounge area unfurls beneath a charcoal-stained, tongue-and-groove ceiling supported by three slender square columns, each framing a view of leafy gum trees beyond a glass-paneled balustrade. Center stage is a low-profile, three-seater sofa upholstered in cool gray fabric, its woven rope–textured arms cradling three plump charcoal cushions. Flanking the sofa on the left and right are two generous, gently curved lounge chairs in matching gray, each paired with a single emerald-green throw pillow; in front of the sofa sits a rectangular coffee table with a pale wooden top and slim black metal legs, supporting a squat brass planter filled with succulent greenery. Scattered around this grouping are three sculptural side tables in minty teal—two hourglass-shaped pedestals and one round-topped stool—providing additional surfaces for a black ceramic mug and decorative tray. Underfoot, an expansive area rug weaves abstract blocks of jade, slate and ivory, echoing the verdant potted specimens that occupy a tiered set of five metallic gold wall shelves to the right and a tall planter beside the sofa. To the far right, a textured stone hearth anchors a recessed fireplace, its natural rock façade punctuating the otherwise streamlined charcoal-clad walls. Every element—from the soft curves of the seating to the rich interplay of greens, grays and natural stone—creates a harmonious indoor-outdoor retreat.

Figure 17: Each row shows a densely captioned image-text pair from our *Lens-800M* training set.