

Boogu-Image-0.1: Boosting Open-Source Unified Multimodal Understanding and Generation

Boogu Team*

Abstract

We introduce Boogu-Image-0.1, an open-source unified multimodal understanding and generation model family, comprising Base, Turbo, Edit, and Edit-Turbo variants. It delivers competitive performance in high-quality text-to-image generation, fast inference, instruction-based editing, and bilingual (Chinese-English) text rendering. Closed-source multimodal systems like Nano-Banana-Pro and GPT-Image-2 achieve strong performance through system-level integration rather than a single model, yet their internal practices remain largely undisclosed. In this work, we demonstrate that targeted improvements in model understanding, data quality, and training pipelines, coupled with agentic inference-time scaling, can substantially enhance generation and editing performance even under highly constrained compute budgets. Comprehensive evaluations show that Boogu-Image-0.1 consistently matches or surpasses other open-source models across standard benchmarks, and achieves results approaching leading closed-source systems. Notably, this is accomplished with only 208.62 million unique images. The base model’s theoretical training cost is only approximately \$400K. We share practical discussions that we believe are valuable to the broader research community, and release weights, code, and recipes under Apache 2.0 to advance the open ecosystem for unified multimodal understanding and generation. Our code is available here: <https://github.com/Boogu-Project/Boogu-Image>.

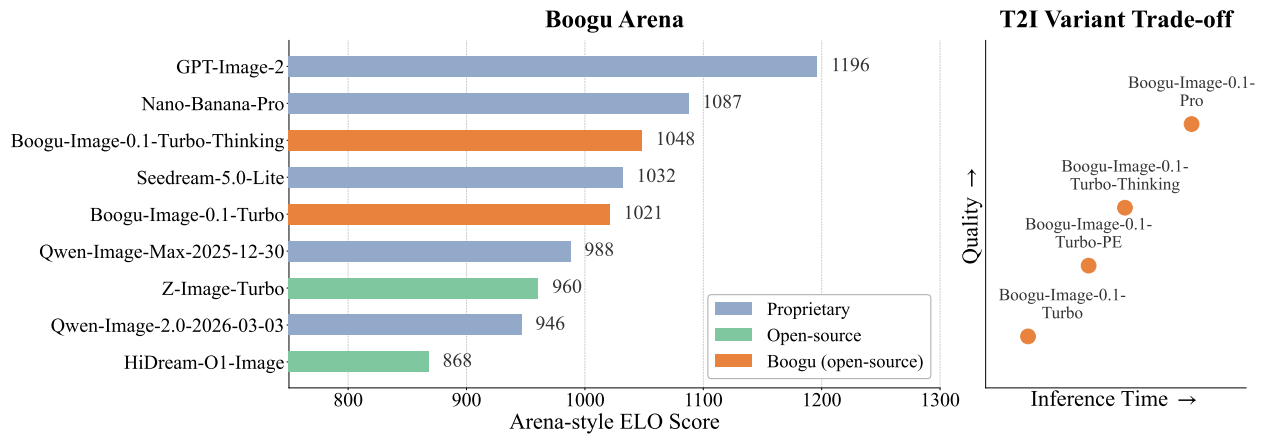


Figure 1 Left: Performance comparison on Boogu Arena. Boogu-Image-0.1 achieves top-tier results, outperforming other open-source counterparts as of June 15, 2026. The details of Boogu Arena are provided in Sec 2.2.1. **Right: Inference time and performance of Boogu-Image variants (illustrative only).** Scaling the inference time leads to higher text-to-image (T2I) generation quality, as illustrated in the figure.

*Detailed author information can be found in Appendix A.



Figure 2 Boogu-Image-0.1 generates diverse, high-fidelity images across distinct visual domains within a single unified model. From left to right: **(a)** a photorealistic black-and-white portrait demonstrating fine-grained facial details and film-like tonal rendering; **(b)** a Chinese ink-and-gold landscape painting reflecting culturally grounded stylistic generation; **(c)** a typographic poster rendering Chinese calligraphy “A single spark can start a prairie fire” with accurate stroke structure and layout; **(d)** a 3D miniature scene of a glowing sapling on a hill, illustrating compositional reasoning and spatial coherence; and **(e)** a clean text-rendering result presenting our paper title “Boogu-Image-0.1: Boosting Open-Source Unified Multimodal Understanding and Generation”.



Figure 3 Boogu’s qualitative results on text-rendering. Boogu-Image-0.1 can accurately render English and Chinese texts with coherent typography and delicate layout design.



Figure 4 Boogu’s qualitative results on photography and style transfer. Boogu-Image-0.1 generates highly photorealistic images and renders diverse artistic styles across distinct visual domains.

Contents

	Preamble	
1	Introduction	6
	Evaluation	
2	Experimental Results	7
2.1	Rethinking Open-Source Research Evaluation	7
2.2	Text-to-Image Generation	8
2.2.1	Boogu Arena Benchmark	8
2.2.2	Qwen-Image Benchmark	9
2.2.3	LongText Benchmark	11
2.2.4	Benchmarks We Do Not Use	12
2.2.5	Qualitative Comparison	13
2.3	Image-to-Image Generation	19
2.3.1	ImgEdit Benchmark	19
2.3.2	Qualitative Comparison	20
3	Experimental Analysis and Discussion	24
3.1	Boosting Generation via Better Understanding	24
3.1.1	Instruction Encoder: The Sensor of Text-to-Image Models	25
3.1.2	Prompt Rewriter: A Translator, Not an Enhancer	26
3.1.3	Captions: Good Supervision Requires Understanding User Demand	28
3.1.4	Model Router: The Right Model for the Right Task	28
3.2	General Discussion on Boogu Experiments	30
3.2.1	Discussions Beyond Leaderboards	30
3.2.2	Data Ablation Evaluation: Just Enough Data Is Not Enough	33
3.2.3	Ablation Experiments on Multimodal Learning and Understanding	40
4	Conclusion, Limitations, and Future Directions	45
	Appendix	
A	Contributors	55
B	Background and Technical Choices	56
C	Boosted Orthogonal Guidance	64

Preamble

1 Introduction

Recent closed-source systems such as the GPT-Image series [93, 94], the Nano-Banana series [43, 44], and the Seedream series [14, 15, 113] have advanced at a striking pace: they not only render fine-grained visual details with remarkable fidelity, but also reason over complex user instructions and, in some cases, augment their world knowledge through web search at generation time. Open-source efforts such as FLUX [7, 8], Ideogram [2], Qwen-Image [130, 143], Z-Image [138], and Hunyuan-Image [19] have made tremendous progress in closing this gap, primarily by improving visual quality and by attaching an auxiliary LLM to rewrite user instructions. To push the boundaries of open-source models, we treat understanding as a first-class design target alongside data quality, the training recipe, and the inference pipeline. Results show that coordinated improvements across these three axes yield substantial gains in instruction following and generation fidelity.

Our emphasis on understanding stems from a conviction that image generation has entered a new era, evolving from Text-to-Image toward Requirement-to-Image Generation. Users are no longer satisfied with a single descriptive prompt; they expect models to interpret complex intentions, implicit constraints, multi-level instructions, and cross-modal contextual cues. Moreover, real-world requirements are highly diverse: they range from simple to intricate and vary in cost and latency expectations, so a single model configuration rarely serves them all equally well. Understanding is the critical bridge connecting ambiguous human requirements to precise visual generation, and to the appropriate strategy under each user’s constraints.

To realize this understanding, we decompose it into system components, beginning with how the model perceives language. We regard the text encoder as the sensor for language: a sufficiently capable text encoder is indispensable for interpreting users’ requirements. To this end, we systematically analyze models of varying scales and confirm that stronger LLMs yield stronger text-encoding capabilities—a trend noted, though not systematically studied, in prior works [22, 133, 143]. Guided by this finding, we adopt Qwen3-VL-8B [3], balancing text-encoding capability against a parameter budget acceptable to the open-source community.

We further employ Agentic Image Generation, which injects understanding into generation at inference time [60, 142]. Instead of treating the model as a passive text-to-image mapping, we wrap it in an agent that interprets the user’s request and orchestrates generation accordingly. First, the agent performs agentic prompt rewriting [60, 81], refining the prompt to better capture the intended meaning. Second, it selects among different model variants (e.g., Base vs. Turbo) to balance speed and quality across diverse scenarios. Third, it incorporates inference-time techniques such as Reflection [75] to further improve quality. As shown in Figure 1, model performance improves progressively as more understanding ability is injected.

Beyond understanding, training efficiency is a core design principle. Under a constrained compute budget, we design a data curation pipeline that prioritizes quality over quantity with various training optimization strategies. This allows us to train a text-to-image base model from scratch using only 208.62 million unique images, at a total cost of roughly \$400K, while achieving performance competitive with the state of the art.

Equally important, we find that many seemingly secondary details are in fact decisive in practice, including evaluation protocols [42, 57], data filtering [130], and caption design [138]. Indeed, their opacity is itself a major barrier to open-source progress: they are often learned through costly trial and error within industrial teams, yet rarely described in public technical reports. By documenting them, we aim to reduce the hidden engineering cost for the open-source community and provide practical reference points for future work.

Building on these designs, we present Boogu-Image-0.1, an open-source system that unifies understanding and generation. By promoting understanding to a first-class component, it moves beyond conventional Text-to-Image pipelines toward the Requirement-to-Image paradigm, faithfully translating complex user intents into high-fidelity images. It achieves top-tier performance on Boogu Arena and state-of-the-art results among open-source models on the recently released Qwen-Image-Bench [73]. We release the model along with our findings and practical lessons to support future research on understanding-driven generation. We hope Boogu-Image-0.1 marks a step toward image generation systems that truly understand what users want.

Evaluation

2 Experimental Results

2.1 Rethinking Open-Source Research Evaluation

Public benchmarks remain valuable [21, 35, 62, 91, 136], but only when used and interpreted correctly. The reality is that few state-of-the-art closed-source models report results on GenEval [42] or DPG-Bench [57], even though these models are widely recognized as far stronger than most open-source ones. To illustrate this, we compare six recent text-to-image models on their human-preference ranking (LMArena Elo) versus their scores on GenEval and DPG-Bench (Figure 5). If these benchmarks tracked real capability, stronger models should score higher; instead, we observe clear rank inversions. Most strikingly, GPT-Image-2 [94], the strongest by human preference, ranks only mid-pack on both benchmarks. We observe the same inversions in our own models: on GenEval [42], a SANA-VAE configuration of our model reaches 0.92, yet our deployed and stronger model scores only 0.85. Consequently, these benchmarks no longer reliably correlate with model capability. We attribute this to three causes.

- **Many public benchmarks do not reflect real application scenarios.** They assess isolated capabilities under constrained, synthetic conditions that diverge substantially from production usage. As a result, a model that surpasses Nano-Banana-Pro [43] across academic benchmarks may still deliver a markedly worse user experience in deployment—a misalignment between the distributions sampled by static benchmarks and those encountered in practice. Evaluation should be derived from target application scenarios rather than treated as an independent objective: the question is not how a model scores on a benchmark, but whether a defined user requirement is met and how it can be measured directly.
- **Most widely adopted benchmarks have approached saturation.** State-of-the-art models routinely achieve near-ceiling scores, compressing the dynamic range for comparison and rendering small score differences statistically uninformative. In this regime, observed gaps are driven by variance, prompt formatting, and benchmark-specific overfitting rather than genuine capability differences.
- **Data contamination and test-set leakage are pervasive.** Large-scale pre-training corpora frequently incorporate public evaluation sets, so reported scores may reflect memorization rather than generalization. Inconsistent reporting standards compound the problem: some results are obtained under training conditions that include leaked evaluation data, undermining both the interpretability of individual scores and the validity of cross-model comparisons.

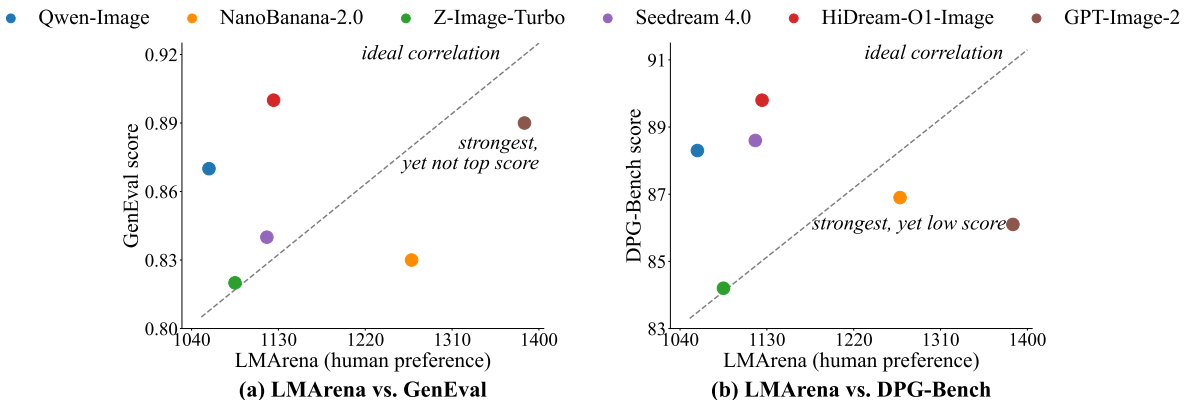


Figure 5 Public benchmarks fail to track human preference. We compare six recent text-to-image models, plotting their LMArena [145] Elo (a human-preference leaderboard reflecting real-world capability) against their (a) GenEval [42] and (b) DPG-Bench [57] scores. The gray dashed line marks the ideal case of perfect rank correlation. Strikingly, the strongest model by human preference, GPT-Image-2, ranks only in the middle of the pack on both benchmarks. These inversions reveal a gap between saturated academic benchmarks and genuine generation quality.

2.2 Text-to-Image Generation

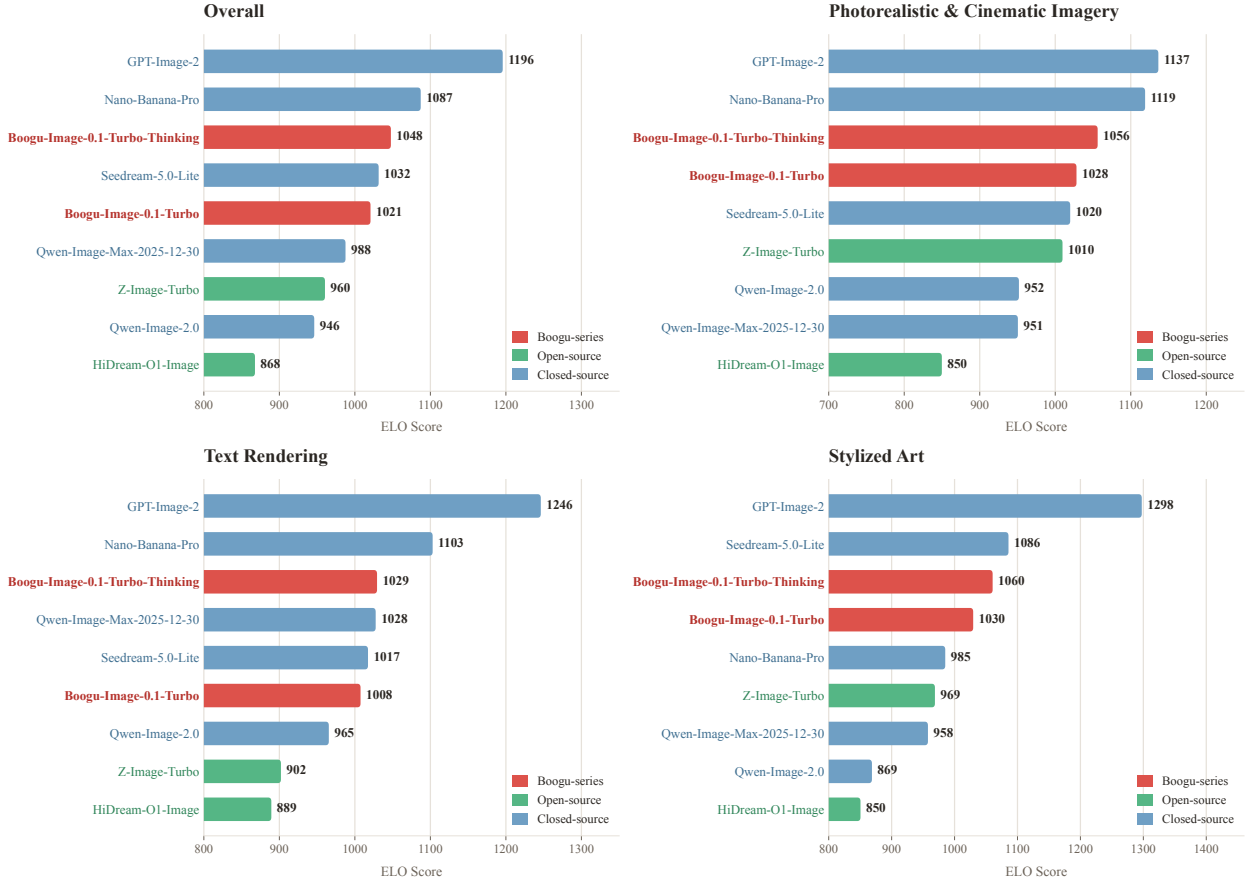


Figure 6 Boogu Arena Text-to-Image Elo scores. Pairwise blind votes are aggregated into Elo ratings, where higher scores indicate stronger relative human preference. The four panels display the overall ratings alongside three specific categories. Across all categories, Boogu models lead the open-source tier, trailing only the closed-source frontier models (GPT-Image-2 [94] and Nano-Banana-Pro [43]). Notably, the thinking variant yields the most substantial gains in the prompt-sensitive text-rendering category. Note: Qwen-Image-2.0 is Qwen-Image-2.0-2026-03-03.

2.2.1 Boogu Arena Benchmark

Since we do not have access to the *arena.ai* [1] evaluation platform, we introduce the Boogu Arena to ensure a fair comparison to the greatest extent possible. Specifically, we built an in-house benchmark that faithfully emulates the original Arena pipeline (see Figure 7), including its prompt design and blind pairwise battles.

Prompt design. We organize the prompt space into three core application categories: photorealistic and cinematic imagery, text rendering, and stylized art. Each category is populated with roughly 400 fine-grained sub-scene keywords. To better reflect real-world user behavior, a powerful VLM further instantiates each prompt along two axes. The first is prompt length, spanning short, medium, and long at a 3:4:3 ratio. The second is user persona, covering 27 roles across novice, intermediate, and professional tiers at a 5:3:2 ratio. The two axes are coupled rather than drawn independently. The long paragraph-style prompts, for instance, are never paired with novice personas. This procedure yields 1,200 bilingual prompts in total. Each prompt is authored in both Chinese and English, and a single language is randomly selected at test time and shared across all compared models. To ensure evaluation fairness, each model is evaluated only once per prompt using a random seed. We will release the designed prompts.

Blind battle. We conduct a strictly blind pairwise evaluation. For every prompt we present the outputs of two anonymized models side by side, the annotator votes among “A is better”, “B is better”, “tie good”, and “tie bad”, and the model identities are revealed only after the vote is cast. Sample selection and model pairing both follow a weighted inverse-frequency strategy so that every prompt and every model receives sufficient coverage, and the collected votes are aggregated into Elo scores via a Bradley-Terry model.

Following this evaluation protocol, we compare our Boogu models (Boogu-Image-0.1-Turbo and Boogu-Image-0.1-Turbo-Thinking) against two open-source baselines (Z-Image-Turbo [138] and HiDream-O1-Image [17]) and five closed-source counterparts (GPT-Image-2 [94], Nano-Banana-Pro [43], Seedream-5.0-Lite [15], Qwen-Image-Max-2025-12-30 [99], and Qwen-Image-2.0-2026-03-03 [143]). In total, we collected over 4,000 votes from the pairwise battles conducted among these nine models.

Figure 6 details the overall and per-category Elo scores of the nine evaluated models. Boogu models consistently dominate the open-source tier across all categories, surpassed only by top closed-source models (GPT-Image-2 [94] and Nano-Banana-Pro [43]). This advantage is most evident in photorealistic and stylized content, where Boogu-Image-0.1-Turbo-Thinking substantially closes the gap with the closed-source frontier. In text rendering, the thinking variant clearly outperforms its non-thinking counterpart. We attribute this to the high prompt sensitivity of text generation: thinking-based prompt enhancement explicitly includes the exact text to be rendered, producing better-structured prompts that improve text correctness.

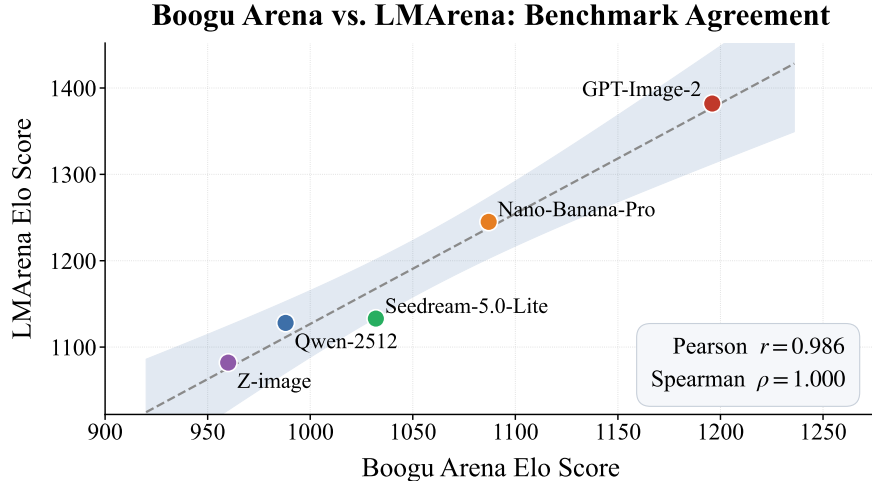


Figure 7 Agreement between Boogu Arena and LMArena Elo ratings. Each point corresponds to one text-to-image model, with its Elo score on Boogu Arena (x-axis) plotted against its score on the public LMArena leaderboard (y-axis). The dashed line shows a linear fit with the shaded region indicating the 95% confidence band. The two benchmarks exhibit strong agreement, with a Pearson correlation of $r = 0.986$ and a perfect Spearman rank correlation of $\rho = 1.0$, indicating that Boogu Arena faithfully reproduces the model ranking established by large-scale public human-preference voting.

2.2.2 Qwen-Image Benchmark

Qwen-Image-Bench [73] is a recently released, high-quality benchmark, published after we froze our text-to-image development. Because it is less affected by common issues in long-standing benchmarks such as data leakage, we regard it as a reliable testbed for modern image generation models. As shown in Tables 1 and 2, Boogu-Image-0.1-Base-Thinking achieves the best overall score among all evaluated open-source models under both Chinese (53.57) and English (53.73) prompts, surpassing strong open-source baselines such as Qwen-Image-2512 [99] and HunyuanImage-3.0 [19]. The improvement from the thinking variant is most pronounced in the Creativity dimension, where Base-Thinking gains over 8 points over its non-thinking counterpart (e.g., 48.62 $\rightarrow$ 56.74 in Chinese), consistent with our understanding-driven design. These trends are stable across both languages, indicating robust bilingual capability. Due to time constraints, the evaluation does not yet cover all available open-source baselines.

Model	License	Quality	Aesthetics	Alignment	Realism	Creativity	Overall $\uparrow$
GPT-Image-2 [94]	Closed	58.65	67.53	65.85	57.38	75.23	64.69
Nano-Banana-2.0 [44]	Closed	54.77	<u>61.08</u>	<u>62.40</u>	<u>54.28</u>	<u>67.05</u>	<u>59.82</u>
Nano-Banana-Pro [43]	Closed	<u>55.67</u>	60.26	61.25	54.07	66.23	59.45
Qwen-Image-2.0-Pro [143]	Closed	54.39	58.67	59.28	51.83	64.94	57.84
Seedream-5.0-Lite [15]	Closed	52.55	58.40	58.90	51.92	65.29	57.22
FLUX-2-Max [9]	Closed	53.64	56.85	57.35	49.35	56.50	55.33
GPT-Image-1 [93]	Closed	52.34	55.09	56.28	48.14	55.78	54.07
Boogu-Image-0.1-Base-Thinking	Apache-2.0	50.58	55.20	55.99	47.35	56.74	53.57
Boogu-Image-0.1-Turbo-Thinking	Apache-2.0	50.89	54.36	56.28	47.67	53.33	53.13
Qwen-Image-2512 [99]	Apache-2.0	51.76	54.74	52.72	47.00	50.19	52.06
Imagen-4.0-Ultra [45]	Closed	50.90	54.25	54.02	45.59	51.14	51.99
Boogu-Image-0.1-Turbo	Apache-2.0	51.24	53.50	53.54	46.11	48.91	51.53
Boogu-Image-0.1-Base	Apache-2.0	50.41	53.14	52.91	45.42	48.62	50.96
HunyuanImage-3.0 [19]	Other	50.35	53.57	52.00	44.31	49.12	50.81
Imagen-4.0 [45]	Closed	50.16	52.68	51.64	44.84	47.94	50.29
Qwen-Image [130]	Apache-2.0	48.44	52.25	50.72	43.16	47.30	49.23
Kling-Image-2.1 [70]	Closed	49.11	50.15	49.18	44.74	44.67	48.26
GLM-Image [139]	Apache-2.0	49.26	50.64	47.90	44.69	45.23	48.19

Table 1 Our model series achieves top-tier performance among the evaluated open-source models on Qwen-Image-Bench (Chinese prompts). Gray shading denotes closed-source models, and our models are highlighted in **red**. Best scores are **bolded** and second-best scores are underlined. Results for other methods are taken from [73].

Model	License	Quality	Aesthetics	Alignment	Realism	Creativity	Overall $\uparrow$
GPT-Image-2 [94]	Closed	59.09	68.48	65.78	59.40	75.34	65.23
Nano-Banana-2.0 [44]	Closed	54.86	62.63	61.11	54.66	64.49	59.59
Nano-Banana-Pro [43]	Closed	55.30	61.38	60.30	<u>55.91</u>	64.54	59.33
Qwen-Image-2.0-Pro [143]	Closed	55.16	60.36	57.86	53.06	63.59	57.90
Seedream-5.0-Lite [15]	Closed	54.01	59.96	58.63	53.86	63.64	57.80
FLUX-2-Max [9]	Closed	53.99	58.77	57.31	50.69	57.79	56.14
GPT-Image-1 [93]	Closed	52.50	55.77	55.35	48.25	57.29	54.24
Boogu-Image-0.1-Base-Thinking	Apache-2.0	51.52	55.89	55.58	47.23	56.24	53.73
Boogu-Image-0.1-Turbo-Thinking	Apache-2.0	51.92	56.28	55.67	47.89	52.70	53.58
Imagen-4.0-Ultra [45]	Closed	51.16	55.64	53.75	46.00	51.32	52.42
Boogu-Image-0.1-Turbo	Apache-2.0	51.30	53.91	53.37	46.52	48.90	51.61
HunyuanImage-3.0 [19]	Other	50.76	54.66	53.16	45.33	48.33	51.35
Qwen-Image-2512 [99]	Apache-2.0	51.84	54.40	51.44	47.80	47.75	51.32
Imagen-4.0 [45]	Closed	50.63	53.93	52.56	45.13	48.56	51.08
Boogu-Image-0.1-Base	Apache-2.0	50.38	53.90	52.62	46.54	47.52	51.00
Kling-Image-2.1 [70]	Closed	49.04	50.94	50.47	44.29	46.23	48.89
Qwen-Image [130]	Apache-2.0	48.45	51.18	50.04	43.45	45.37	48.48
GLM-Image [139]	Apache-2.0	49.86	49.98	47.49	44.25	44.67	47.86

Table 2 Our model series achieves top-tier performance among the evaluated open-source models on Qwen-Image-Bench (English prompts). Gray shading denotes closed-source models, and our models are highlighted in **red**. Best scores are **bolded** and second-best scores are underlined. Results for other methods are taken from [73].

2.2.3 LongText Benchmark

LongText-Bench [41] is a benchmark for evaluating long-form text rendering in image generation, a capability that is increasingly important for practical applications such as posters, slides, documents, comics, and instruction-heavy visual designs. Similar to Qwen-Image-Bench [73], it adopts a model-based annotation protocol that assesses not only OCR-level correctness, but also readability, layout consistency, and whether the generated image preserves the intended textual content and visual structure.

As shown in Table 3, our models achieve top-tier text-rendering performance among the compared open-source models, and are particularly strong on Chinese (ZH) prompts, where Boogu-Image-0.1-Turbo-Thinking ranks second overall (0.985), trailing only the closed-source Seedream-4.5. We note, however, that this benchmark primarily measures OCR-level text accuracy on relatively short sequences (fewer than 100 characters) and does not fully capture overall visual quality in dense layouts. Under such dense-text settings, we observe that the Base model produces more coherent typography than Turbo, whereas Turbo, despite scoring comparably on this benchmark, tends to introduce visible artifacts in densely packed regions. Due to time constraints, the evaluation does not yet cover all available open-source baselines.

Limitations of LongText-Bench. While useful, this benchmark has two notable limitations. First, its model-based protocol primarily scores whether the target characters are correctly rendered (i.e., OCR-level accuracy), but does not assess the fidelity of the rendered text, such as whether the glyphs are visually natural, well-integrated with the surrounding layout and materials, and free of artifacts. As a result, a model can attain a high score even when the rendered text exhibits noticeable visual flaws. Second, the benchmark is largely saturated: apart from a single outlier, the top-ranked models are tightly clustered within a narrow range (roughly 0.96–0.99 in average score), leaving little headroom to discriminate among strong models. We therefore view LongText-Bench as a reliable check for text correctness, but recommend complementing it with human evaluation or perceptual metrics when assessing the overall quality of long-form text rendering.

Model	Open Source	AVG ↑	EN ↑	ZH ↑
Seedream-4.5 [14]	Closed	0.988	0.989	0.987
HiDream-O1-Image [17]	MIT	<u>0.979</u>	0.979	0.978
ERNIE-Image-PE [81]	Apache-2.0	0.973	<u>0.980</u>	0.966
Boogu-Image-0.1-Turbo-Thinking	Apache-2.0	0.971	0.957	0.985
GLM-Image [139]	Apache-2.0	0.966	0.952	0.979
Nano-Banana-2.0 [44]	Closed	0.965	<u>0.981</u>	0.949
JoyAI-Image [118]	Apache-2.0	0.963	0.963	0.963
Seedream-5.0-Lite [15]	Closed	0.962	0.961	0.963
Qwen-Image-2512 [99]	Apache-2.0	0.961	0.956	0.965
GPT-Image-2 [94]	Closed	0.961	0.960	0.961
Boogu-Image-0.1-Turbo	Apache-2.0	0.961	0.944	0.977
Boogu-Image-0.1-Base	Apache-2.0	0.961	0.952	0.969
Qwen-Image [130]	Apache-2.0	0.945	0.943	0.946
Seedream-4.0 [113]	Closed	0.924	0.921	0.926
Z-Image-Turbo [138]	Apache-2.0	0.922	0.917	0.926
GPT-Image-1 [High] [93]	Closed	0.788	0.956	0.619

Table 3 Our model series achieves top-tier performance among the evaluated models on LongText-Bench.

Models are sorted by average score. Gray shading denotes closed-source models, and our models are highlighted in **red**. Best scores are **bolded** and second-best scores are underlined. Results for other methods are taken from [17, 139]. Actually, Base model outperforms Turbo in dense text rendering. While Turbo introduces noticeable visual artifacts in dense layouts, it still scores high on the LongText benchmark, as the metric primarily evaluates text accuracy for shorter sequences (<100 characters) rather than overall visual quality.

2.2.4 Benchmarks We Do Not Use

We observe that a significant gap still remains between some text-to-image benchmarks (e.g., GenEval [42] and DPG-Bench [57]) and real human preferences, as discussed in Figure 5. Consequently, we do not adopt these benchmarks for our primary evaluation. We nonetheless report the corresponding results as a supplementary reference for historical continuity (Table 4 for GenEval and Table 5 for DPG-Bench), and we suggest that their suitability as primary benchmarks going forward be carefully reconsidered.

Model	License	Single Obj.	Two Obj.	Counting	Colors	Position	Attribute	Overall↑
HiDream-O1-Image [17]	Apache-2.0	1.00	0.99	0.79	0.89	0.93	0.78	0.90
GPT-Image-2 [94]	Closed	<u>0.99</u>	<u>0.98</u>	0.85	0.93	0.85	0.77	<u>0.89</u>
FLUX.2 [Dev] [8]	Non-Comm.	1.00	0.99	0.79	0.93	0.73	<u>0.78</u>	0.87
Qwen-Image [130]	Apache-2.0	<u>0.99</u>	0.92	<u>0.89</u>	0.88	0.76	0.77	0.87
Boogu-Image-0.1-Base	Apache-2.0	<u>0.99</u>	0.95	0.80	0.84	0.85	0.68	0.85
Boogu-Image-0.1-Turbo	Apache-2.0	1.00	0.97	0.85	0.76	0.86	0.60	0.84
Boogu-Image-0.1-Turbo-Think.	Apache-2.0	1.00	0.94	0.88	0.83	0.70	0.68	0.84
Seedream-4.0 [113]	Closed	1.00	0.92	0.71	0.93	0.78	0.68	0.84
Seedream-3.0 [39]	Closed	<u>0.99</u>	0.96	0.91	0.93	0.47	0.80	0.84
GPT-Image-1 [High] [93]	Closed	<u>0.99</u>	0.92	0.85	<u>0.92</u>	0.75	0.61	0.84
Z-Image [138]	Apache-2.0	1.00	0.94	0.78	0.93	0.62	0.77	0.84
Nano-Banana-2.0 [44]	Closed	1.00	0.96	0.71	0.84	<u>0.86</u>	0.65	0.83
Z-Image-Turbo [138]	Apache-2.0	1.00	0.95	0.77	0.89	0.65	0.68	0.82
FLUX.1 [Dev] [7]	Non-Comm.	0.98	0.81	0.74	0.79	0.22	0.45	0.66

Table 4 Performance of the Boogu series and baseline models on GenEval. Closed-source models are shaded in gray. Boogu series models are highlighted in **red**. Best scores in each column are **bolded** and second-best scores are underlined. Models are sorted by overall score in descending order. Results for other methods are taken from [17]. We report these results for reference only, as GenEval exhibits a substantial gap from human preference.

Model	License	Global	Entity	Attribute	Relation	Other	Overall↑
HiDream-O1-Image [17]	Apache-2.0	95.15	92.32	93.74	92.88	90.25	89.83
Seedream-4.5 [14]	Closed	89.24	94.30	92.14	92.23	<u>93.83</u>	<u>88.63</u>
Seedream-4.0 [113]	Closed	93.86	92.24	90.74	<u>93.87</u>	94.16	88.54
Boogu-Image-0.1-Turbo	Apache-2.0	88.54	91.67	92.19	93.20	93.83	88.35
Qwen-Image [130]	Apache-2.0	91.32	91.56	92.02	94.31	92.73	88.32
Seedream-3.0 [39]	Closed	<u>94.31</u>	<u>92.65</u>	91.36	92.78	88.24	88.27
Z-Image [138]	Apache-2.0	93.39	91.22	93.16	92.22	91.52	88.14
FLUX.2 [Dev] [8]	Non-Commercial	92.20	91.36	<u>93.28</u>	93.52	89.72	87.57
Boogu-Image-0.1-Turbo-Thinking	Apache-2.0	90.26	92.03	91.27	92.35	91.98	87.23
Qwen-Image-2512 [99]	Apache-2.0	89.04	91.91	92.39	90.85	93.07	87.20
Boogu-Image-0.1-Base	Apache-2.0	89.33	90.64	92.22	93.72	93.32	87.13
Nano-Banana-2.0 [44]	Closed	85.17	92.55	91.16	90.45	91.08	86.90
GPT-Image-2 [94]	Closed	87.27	91.91	90.85	91.59	91.58	85.98
GPT-Image-1 [High] [93]	Closed	88.89	88.94	89.84	92.63	90.96	85.15
Z-Image-Turbo [138]	Apache-2.0	91.29	89.59	90.14	92.16	88.68	84.86
GLM-Image [139]	Apache-2.0	87.74	90.25	89.08	92.15	90.17	84.78
FLUX.1 [Dev] [7]	Non-Commercial	74.35	90.00	88.96	90.87	88.33	83.84

Table 5 Performance of the Boogu series and baseline models on DPG-Bench. Closed-source models are shaded in gray. Boogu series models are highlighted in **red**. Best scores in each column are **bolded** and second-best scores are underlined. Models are sorted by overall score in descending order. Results for other methods are taken from [17]. We report these results for reference only, as DPG-Bench exhibits a substantial gap from human preference.

2.2.5 Qualitative Comparison

We provide qualitative comparison samples to offer readers a direct visual reference beyond aggregate benchmark scores. The testing samples are sourced from Boogu Arena, so that each compared model is represented by its strongest publicly accessible outputs under the deployed evaluation setting. This setup helps avoid underestimating baselines due to suboptimal local inference configurations, prompt formatting, or sampling choices. Across the selected examples, we compare Boogu-Image-0.1 with strong closed-source models (including Qwen-Image-Max-2025-12-30 [99], Qwen-Image-2.0-2026-03-03 [143], Seedream-5-Lite [15], Nano-Banana-Pro [43], and GPT-Image-2 [94]) and open-source models (Z-Image-Turbo [138] and HiDream-O1 [17]), focusing on visual fidelity, prompt alignment, composition, and aesthetic quality. These examples complement the quantitative evaluations and illustrate how Boogu-Image-0.1 performs in user-facing comparisons.

Specifically, Figures 8–10 present qualitative comparisons for the three categories in Boogu Arena, namely photorealistic and cinematic imagery, stylized art, and text rendering. These visual results demonstrate that Boogu-Image-0.1-Turbo achieves highly competitive performance in prompt alignment, visual quality, and text-rendering fidelity. Notably, it outperforms existing open-source alternatives and effectively narrows the performance gap toward top-tier closed-source models (e.g., Nano-Banana-Pro [43] and GPT-Image-2 [94]), as validated by the human Elo ratings (Figure 6).

In terms of text rendering (Figure 10), Boogu-Image-0.1-Turbo reliably supports precise character-level rendering for short-to-medium text (up to 100 characters for Chinese / 100 words for English). For more challenging scenarios involving ultra-dense text (exceeding 100 characters or words), we recommend deploying Boogu-Image-0.1-Base, which is explicitly optimized for dense text rendering at high (2K) resolution. As shown in Figures 11 and 12, Boogu-Image-0.1-Base remains stable on dense text rendering even at minuscule font sizes, while maintaining rigorous character accuracy and strong aesthetic appeal for practical applications.



Figure 8 Qualitative results from Boogu Arena on photorealistic and cinematic images. Our model can produce photorealistic and cinematic images with superior visual quality.



Figure 9 Qualitative results from Boogu Arena on stylized art. Our model enables the generation of diverse artistic images with superior visual quality.



Figure 10 Qualitative results from Boogu Arena on simple text rendering. Our **Boogu-Image-0.1-Turbo** can precisely renders short-to-medium text (up to 100 characters/words for Chinese/English) into the generated images. While most baseline models accurately render the requested text, our model achieves superior prompt-image alignment and higher visual quality compared to existing open-source counterparts. For dense text rendering (>100 characters/words), we highly recommend utilizing Boogu-Image-0.1-Base tailored for high-fidelity dense text rendering under 2K resolution (see Figure 11 & 12).

This is a portrait-oriented electric bicycle parameter comparison advertising poster (2:3 ratio, 1600×2528 pixels) with a clean, modern aesthetic built on a dark charcoal and vibrant green colour palette against a white background, conveying both technology and eco-friendliness. The layout is divided into five horizontal zones: a brand tagline hero area at the top, a head-to-head visual comparison section, a specifications table area, an application-scenario grid, and a call-to-action footer banner. Typography is primarily sans-serif with a clear hierarchy from bold large headlines to compact body text. In the top-left corner, a green pill-shaped badge contains the white sans-serif text "Longer Range - Smarter Tech - Better Value". Below, the primary headline reads "VOLT X Pro vs VOLT X" in large, bold, dark charcoal sans-serif letters, followed by "Side-by-Side Comparison" in a smaller, lighter font. Below the headline, the text "Discover which model fits your commute — range, power, and features at a glance" is displayed in a medium dark-grey subtitle states "Discover which model fits your commute — range, power, and features at a glance". To the right, two high-gloss 3D-rendered e-bikes are displayed side-by-side, the left bike is labeled "X" in white text on a dark overlay and the right bike "X Pro" in white text on a green overlay. Below the hero, three feature highlights sit horizontally: left has the green bold title "Extended Range" with dark grey description "Up to 75 miles"; center has "Fast Charging" with "0-80% in 2.5 hrs"; right has "Smart Connect" with "App-integrated GPS".

Below the hero, three feature highlights sit horizontally: left has the green bold title "Extended Range" with dark grey description "Up to 75 miles"; center has "Fast Charging" with "0-80% in 2.5 hrs"; right has "Smart Connect" with "App-integrated GPS".

Below the hero, three feature highlights sit horizontally: left has the green bold title "Extended Range" with dark grey description "Up to 75 miles"; center has "Fast Charging" with "0-80% in 2.5 hrs"; right has "Smart Connect" with "App-integrated GPS".

Prompt

Longer Range - Smarter Tech - Better Value

VOLT X Pro vs VOLT X Side-by-Side Comparison

Discover which model fits your commute — range, power, and features at a glance



Extended Range
Up to 75 miles

Fast Charging
0-80% in 2.5 hrs

Smart Connect
App-integrated GPS

Specifications Comparison

01	Battery & Range	VOLT X: 48V 12Ah, 45 mi VOLT X Pro: 48V 20Ah, 75 mi
02	Motor & Speed	VOLT X: 500W, 20 mph VOLT X Pro: 750W, 28 mph
03	Frame & Weight	VOLT X: Aluminum, 55 lbs VOLT X Pro: Carbon-mix, 48 lbs
04	Features & Extras	VOLT X: LCD display, rear light VOLT X Pro: Color TFT, integrated dash cam, turn signals

Ideal For Every Rider

Daily Commute
Office, errands, short trips

Weekend Explorer
Trails, parks, scenic routes

Delivery Pro
Food, packages, logistics

Campus Rider
Students, faculty, green travel

Fitness Enthusiast
Pedal-assist training

Your Adventure
Endless possibilities ahead

Order Your VOLT X Pro Today — Ride Farther, Ride Smarter!
Free shipping on all pre-orders. 2-year warranty included.

Nano-Banana-Pro

Longer Range - Smarter Tech - Better Value

VOLT X Pro vs VOLT X Side-by-Side Comparison

Discover which model fits your commute — range, power, and features at a glance



Extended Range
Up to 75 miles

Fast Charging
0-80% in 2.5 hrs

Smart Connect
App-integrated GPS

Specifications Comparison

01	Battery & Range	VOLT X: 48V 12Ah, 45 mi VOLT X Pro: 48V 20Ah, 75 mi
02	Motor & Speed	VOLT X: 500W, 20 mph VOLT X Pro: 750W, 28 mph
03	Frame & Weight	VOLT X: Aluminum, 55 lbs VOLT X Pro: Carbon-mix, 48 lbs

Ideal For Every Rider

Daily Commute
Office, errands, short trips

Weekend Explorer
Trails, parks, scenic routes

Delivery Pro
Food, packages, logistics

Campus Rider
Students, faculty, green travel

Fitness Enthusiast
Pedal-assist training

Your Adventure
Endless possibilities ahead

Order Your VOLT X Pro Today — Ride Farther, Ride Smarter!
Free shipping on all pre-orders. 2-year warranty included.

Qwen-Image-Max-2025-12-30

Longer Range - Smarter Tech - Better Value

VOLT X Pro vs VOLT X Side-by-Side Comparison

Discover which model fits your commute — range, power, and features at a glance



Extended Range
Up to 75 miles

Fast Charging
0-80% in 2.5 hrs

Smart Connect
App-integrated GPS

Specifications Comparison

01	Battery & Range	VOLT X: 48V 12Ah, 45 mi VOLT X Pro: 48V 20Ah, 75 mi
02	Motor & Speed	VOLT X: 500W, 20 mph VOLT X Pro: 750W, 28 mph
03	Frame & Weight	VOLT X: Aluminum, 55 lbs VOLT X Pro: Carbon-mix, 48 lbs
04	Features & Extras	VOLT X: LCD display, rear light VOLT X Pro: Color TFT, integrated dash cam, turn signals

Ideal For Every Rider

Daily Commute
Office, errands, short trips

Weekend Explorer
Trails, parks, scenic routes

Delivery Pro
Food, packages, logistics

Campus Rider
Students, faculty, green travel

Fitness Enthusiast
Pedal-assist training

Your Adventure
Endless possibilities ahead

Order Your VOLT X Pro Today — Ride Farther, Ride Smarter!
Free shipping on all pre-orders. 2-year warranty included.

Boogu-Image-0.1-Base

Longer Range - Smarter Tech - Better Value

VOLT X Pro vs VOLT X Side-by-Side Comparison

Discover which model fits your commute — range, power, and features at a glance



Extended Range
Up to 75 miles

Fast Charging
0-80% in 2.5 hrs

Smart Connect
App-integrated GPS

Specifications Comparison

01	Battery & Range	VOLT X: 48V 12Ah, 45 mi VOLT X Pro: 48V 20Ah, 75 mi
02	Motor & Speed	VOLT X: 500W, 20 mph VOLT X Pro: 750W, 28 mph
03	Frame & Weight	VOLT X: Aluminum, 55 lbs VOLT X Pro: Carbon-mix, 48 lbs
04	Features & Extras	VOLT X: LCD display, rear light VOLT X Pro: Color TFT, integrated dash cam, turn signals

Ideal For Every Rider

Daily Commute
Office, errands, short trips

Weekend Explorer
Trails, parks, scenic routes

Delivery Pro
Food, packages, logistics

Campus Rider
Students, faculty, green travel

Fitness Enthusiast
Pedal-assist training

Your Adventure
Endless possibilities ahead

Order Your VOLT X Pro Today — Ride Farther, Ride Smarter!
Free shipping on all pre-orders. 2-year warranty included.

Seedream-5

Longer Range - Smarter Tech - Better Value

VOLT X Pro vs VOLT X Side-by-Side Comparison

Discover which model fits your commute — range, power, and features at a glance



Extended Range
Up to 75 miles

Fast Charging
0-80% in 2.5 hrs

Smart Connect
App-integrated GPS

Ideifications Comparison

01	Battery & Range	VOLT X: 48V 12Ah, 45 mi VOLT X Pro: 48V 20Ah, 75 mi
02	Motor & Speed	VOLT X: 500W, 20 mph VOLT X Pro: 750W, 28 mph
03	Frame & Weight	VOLT X: Aluminum, 55 lbs VOLT X Pro: Carbon-mix, 48 lbs
04	Features & Extras	VOLT X: LCD display, rear light VOLT X Pro: Color TFT, integrated dash cam, turn signals

Ideal For Every Rider

Daily Commute
Office, errands, short trips

Weekend Explorer
Trails, parks, scenic routes

Delivery Pro
Food, packages, logistics

Campus Rider
Students, faculty, green travel

Fitness Enthusiast
Pedal-assist training

Your Adventure
Endless possibilities ahead

Order Your VOLT X Pro Today — Ride Farther, Ride Smarter!
Free shipping on all pre-orders. 2-year warranty included.

Z-Image-Turbo

Longer Range - Smarter Tech - Better Value

VOLT X Pro vs VOLT X Side-by-Side Comparison

Discover which model fits your commute — range, power, and features at a glance



Extended Range
Up to 75 miles

Fast Charging
0-80% in 2.5 hrs

Smart Connect
App-integrated GPS

Specifications Comparison

01	Battery & Range	VOLT X: 48V 12Ah, 45 mi VOLT X Pro: 48V 20Ah, 75 mi
02	Motor & Speed	VOLT X: 500W, 20 mph VOLT X Pro: 750W, 28 mph
03	Frame & Weight	VOLT X: Aluminum, 55 lbs VOLT X Pro: Carbon-mix, 48 lbs
04	Features & Extras	VOLT X: LCD display, rear light VOLT X Pro: Color TFT, integrated dash cam, turn signals

Ideal For Every Rider

Daily Commute
Office, errands, short trips

Weekend Explorer
Trails, parks, scenic routes

Delivery Pro
Food, packages, logistics

Campus Rider
Students, faculty, green travel

Fitness Enthusiast
Pedal-assist training

Your Adventure
Endless possibilities ahead

Order Your VOLT X Pro Today — Ride Farther, Ride Smarter!
Free shipping on all pre-orders. 2-year warranty included.

GPT-Image-2

Longer Range - Smarter Tech - Better Value

VOLT X Pro vs VOLT X Side-by-Side Comparison

Discover which model fits your commute — range, power, and features at a glance



Extended Range
Up to 75 miles

Fast Charging
0-80% in 2.5 hrs

Smart Connect
App-integrated GPS

Specifications Comparison

01	Battery & Range	VOLT X: 48V 12Ah, 45 mi VOLT X Pro: 48V 20Ah, 75 mi
02	Motor & Speed	VOLT X: 500W, 20 mph VOLT X Pro: 750W, 28 mph
03	Frame & Weight	VOLT X: Aluminum, 55 lbs VOLT X Pro: Carbon-mix, 48 lbs
04	Features & Extras	VOLT X: LCD display, rear light VOLT X Pro: Color TFT, integrated dash cam, turn signals

Ideal For Every Rider

Daily Commute
Office, errands, short trips

Weekend Explorer
Trails, parks, scenic routes

Delivery Pro
Food, packages, logistics

Campus Rider
Students, faculty, green travel

Fitness Enthusiast
Pedal-assist training

Your Adventure
Endless possibilities ahead

Order Your VOLT X Pro Today — Ride Farther, Ride Smarter!
Free shipping on all pre-orders. 2-year warranty included.

Qwen-Image-2.0-2026-03-03

Longer Range - Smarter Tech - Better Value

VOLT X Pro vs VOLT X Side-by-Side Comparison

Discover which model fits your commute — range, power, and features at a glance



Extended Range
Up to 75 miles

Fast Charging
0-80% in 2.5 hrs

Smart Connect
App-integrated GPS

Specifications Comparison

01	Battery & Range	VOLT X: 48V 12Ah, 45 mi VOLT X Pro: 48V 20Ah, 75 mi
02	Motor & Speed	VOLT X: 500W, 20 mph VOLT X Pro: 750W, 28 mph
03	Frame & Weight	VOLT X: Aluminum, 55 lbs VOLT X Pro: Carbon-mix, 48 lbs
04	Features & Extras	VOLT X: LCD display, rear light VOLT X Pro: Color TFT, integrated dash cam, turn signals

Ideal For Every Rider

Daily Commute
Office, errands, short trips

Weekend Explorer
Trails, parks, scenic routes

Delivery Pro
Food, packages, logistics

Campus Rider
Students, faculty, green travel

Fitness Enthusiast
Pedal-assist training

Your Adventure
Endless possibilities ahead

Order Your VOLT X Pro Today — Ride Farther, Ride Smarter!
Free shipping on all pre-orders. 2-year warranty included.

HiDream-01-Image

Figure 11 Qualitative comparison between our Boogu-Image-0.1-Base and existing SOTA models on dense text rendering under 2K resolution.

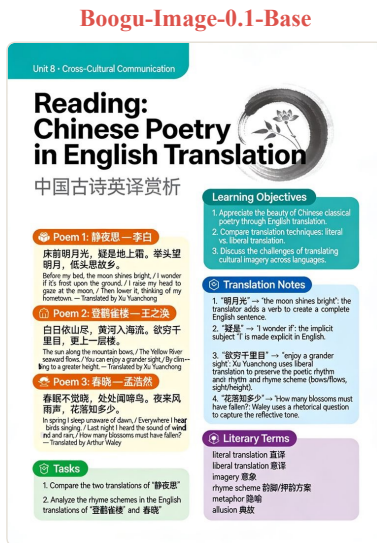
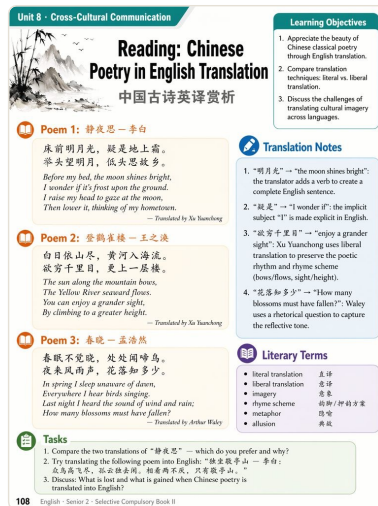
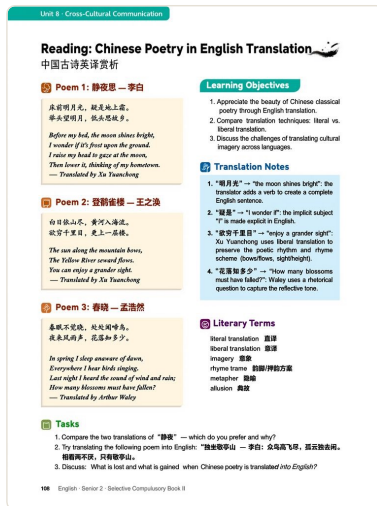


Figure 12 Qualitative comparison between our Boogu-Image-0.1-Base and existing SOTA models on dense text rendering under 2K resolution.

2.3 Image-to-Image Generation

2.3.1 ImgEdit Benchmark

ImgEdit-Bench [136] is a widely adopted framework designed to assess instruction-guided image editing across a diverse range of tasks. We evaluate our Image-to-Image models on ImgEdit-Bench to enable direct comparison with existing methods. As shown in Table 6, Boogu-Image-0.1-Edit-Thinking attains the best overall score (4.64) among all evaluated models, surpassing strong open-source editors such as JoyAI-Image-Edit (4.57) and FireRed-Image-Edit (4.56), as well as leading closed-source systems including Seedream-5.0-Lite (4.42) and Nano-Banana-Pro (4.37). At the sub-task level, our model ranks first on Remove (4.85), Hybrid (4.26), and Action (4.83), and remains highly competitive across the remaining categories, indicating balanced editing capability rather than strength concentrated in a few task types. Notably, the Thinking variant yields consistent gains over the Base variant (4.64 vs. 4.51 overall), with the largest margin on the challenging Extract task (4.32 vs. 3.69); this suggests that explicit reasoning helps the model parse and localize complex editing instructions before generation. We further note that Extract is difficult for essentially all models—several strong closed-source systems score below 3.5 (e.g., Seedream-4.5 at 2.97)—which likely reflects both the intrinsic difficulty of the task and higher variance in VLM-based scoring for it.

Discussion and Limitations. Although Boogu-Image-0.1-Edit-Thinking attains the highest overall score on ImgEdit-Bench, we caution against over-interpreting this ranking. In our own human evaluations, we find that Nano-Banana-Pro [43] outperforms our models on a range of real editing scenarios, despite scoring lower on this benchmark (4.37 vs. 4.64 overall). This inversion highlights a broader issue: ImgEdit-Bench relies primarily on Vision-Language Models (VLMs) as automated evaluators, and current VLMs remain limited in fine-grained visual understanding, making their scores an imperfect proxy for human preference. As a result, the benchmark tends to compress the true quality gap between models and can reward outputs that satisfy instruction-level checks while overlooking subtler perceptual defects.

Moreover, its task taxonomy is relatively coarse: the nine covered categories fall short of capturing the full diversity of real-world editing operations—such as multi-step and multi-region edits, precise text editing, subject-consistent and reference-guided generation, and fine-grained local adjustments—so a high aggregate score does not necessarily imply strong performance on the broader range of edits encountered in practice. We therefore report ImgEdit-Bench results primarily as a supplementary reference for historical continuity, and we suggest that its suitability as a primary benchmark going forward be carefully reconsidered. For assessments that better reflect end-user experience, we place greater weight on human evaluations and qualitative comparisons.

Model	License	Add	Adjust	Extract	Replace	Remove	BG	Style	Hybrid	Action	Overall ↑
Boogu-Image-0.1-Edit-Thinking	Apache-2.0	4.59	4.64	<u>4.32</u>	4.69	4.85	<u>4.60</u>	<u>4.94</u>	4.26	4.83	4.64
JoyAI-Image-Edit [118]	Apache-2.0	4.63	4.52	<u>4.32</u>	4.71	<u>4.76</u>	4.53	4.88	4.09	4.69	<u>4.57</u>
FireRed-Image-Edit [123]	Apache-2.0	4.55	<u>4.66</u>	4.34	<u>4.75</u>	4.58	4.45	4.97	4.07	4.71	4.56
Boogu-Image-0.1-Edit	Apache-2.0	<u>4.71</u>	<u>4.50</u>	3.69	<u>4.65</u>	<u>4.75</u>	<u>4.44</u>	<u>4.94</u>	<u>4.04</u>	4.90	<u>4.51</u>
Qwen-Image-Edit-2511 [101]	Apache-2.0	4.54	4.57	4.13	4.70	4.46	4.36	4.89	<u>4.16</u>	4.81	4.51
LongCat-Image-Edit [122]	MIT	4.44	4.53	3.83	4.80	4.60	4.33	4.92	3.75	4.82	4.45
Seedream-5.0-Lite [15]	Closed	4.93	4.69	3.01	4.41	4.45	4.65	4.93	3.91	4.82	4.42
Nano-Banana-Pro [43]	Closed	4.44	4.62	3.42	4.60	4.63	4.32	4.97	3.64	4.69	4.37
FLUX.2-Dev [9]	Non-Commercial	4.50	4.18	3.83	4.65	4.65	4.31	4.88	3.46	4.70	4.35
Seedream-4.5 [14]	Closed	4.57	4.65	2.97	4.66	4.46	4.37	4.92	3.71	4.56	4.32
Qwen-Image-Edit-2509 [100]	Apache-2.0	4.34	4.27	3.42	4.73	4.36	4.37	4.91	3.56	4.80	4.31
Seedream-4.0 [113]	Closed	4.33	4.38	3.89	4.65	4.57	4.35	4.22	3.71	4.61	4.30
Nano-Banana [43]	Closed	4.62	4.41	3.68	4.34	4.39	4.40	4.18	3.72	<u>4.83</u>	4.29
Step1X-Edit-v1.2 [83]	Apache-2.0	3.91	4.04	2.68	4.48	4.26	3.90	4.82	3.23	4.22	3.95

Table 6 Boogu-Image-0.1-Edit-Thinking achieves the best overall performance on ImgEdit-Bench. Models are sorted by overall score. **BG** denotes the background-change task. Gray shading denotes closed-source models, and our models are highlighted in **red**. Best scores are **bolded** and second-best scores are underlined. Results for other methods are taken from [118]. Note that results are presented for reference only, as VLM evaluations on ImgEdit do not fully align with human preferences.

2.3.2 Qualitative Comparison

We provide qualitative comparison samples to offer readers a direct visual reference that complements the aggregate benchmark scores. The evaluation samples are drawn from the public ImgEdit-Bench [136] as well as our in-house image-to-image benchmark. Similar to the text-to-image evaluation in Section 2.2, the selected samples span three aspects: photorealistic editing, precise scene-text editing from our in-house benchmark, and diverse stylization. We compare Boogu-Image-0.1-Edit against several state-of-the-art image-to-image models, including three closed-source systems—GPT-Image-2 [94], Nano-Banana-Pro [43], and Seedream-5.0-Lite [15]—together with the open-source Qwen-Image-Edit-2511 [101]. Figures 13–15 present the qualitative comparisons for these three aspects, respectively. Inherited from our powerful T2I base model Boogu-Image-0.1-Base, Boogu-Image-0.1-Edit achieves highly competitive image-to-image performance among the compared models, showing particular strength in instruction alignment, photorealism, precise scene-text modification, and artistic stylization.

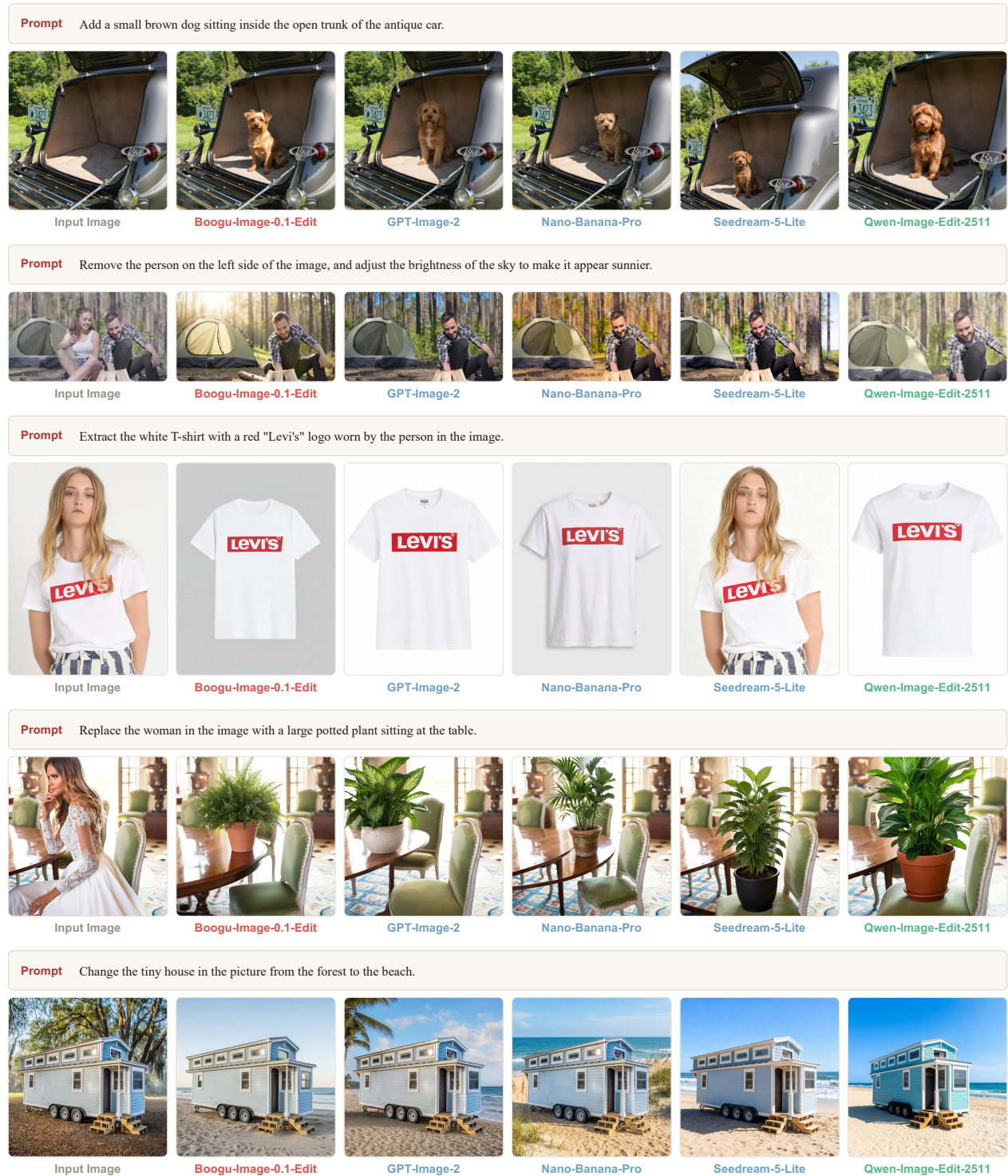
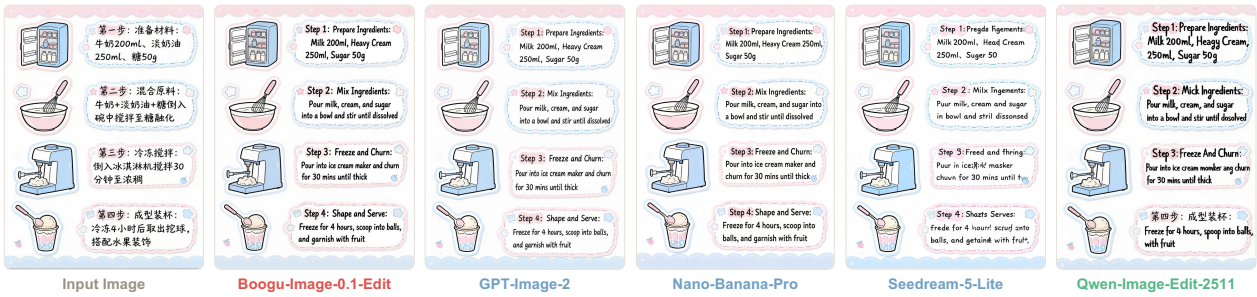


Figure 13 Qualitative comparison on ImgEdit-Bench. Boogu-Image-0.1-Edit can achieve top-tier performance in diverse editing tasks, including but not limited to addition, removal, extraction, and replacement.

Prompt 将挂画上的“人生苦短”替换为“岁月漫长”，并将“咖啡倒满”替换为“清茶一杯”。



Prompt 请将图片中的所有中文文本替换为英文，具体操作如下：将第一步的标题“第一步：准备材料：”替换为“Step 1: Prepare Ingredients:”，内容“牛奶200ml、淡奶油”替换为“Milk 200ml, Heavy Cream”，“250ml、糖50g”替换为“250ml, Sugar 50g”；将第二步的标题“第二步：混合原料：”替换为“Step 2: Mix Ingredients:”，内容“牛奶+淡奶油+糖倒入”替换为“Pour milk, cream, and sugar”，“碗中搅拌至糖融化”替换为“into a bowl and stir until dissolved”；将第三步的标题“第三步：冷冻搅拌：”替换为“Step 3: Freeze and Churn:”，内容“倒入冰淇淋机搅拌30”替换为“Pour into ice cream maker and churn”，“分钟至浓稠”替换为“for 30 mins until thick”；将第四步的标题“第四步：成型装杯：”替换为“Step 4: Shape and Serve:”，内容“冷冻4小时后取出挖球，”替换为“Freeze for 4 hours, scoop into balls,”，“搭配水果装饰”替换为“and garnish with fruit”。



Prompt 请将图片下方的文字“WE INVITE YOU TO\nChristmas\nPARTY”整体向上移动，放置在圣诞树图案的正上方，保持居中对齐。



Prompt 将图片顶部居中位置的“ZENLEAF”修改为水彩笔刷风格。文字应呈现出半透明的绿色水彩质感，边缘有轻微的晕染效果，笔触自然流畅，仿佛是用画笔直接在纸上书写而成，与整体的草本自然主题相呼应。



Figure 14 Qualitative comparison on scene-text editing. Boogu-Image-0.1-Edit enables diverse scene-text editing tasks, including but not limited to text replacement, text translation, text re-organization, and text stylization.

3 Experimental Analysis and Discussion

In this section, we share several key insights and observations encountered during our experiments. While these findings may not seem entirely “novel”, they have clearly not received sufficient attention from the research community, as evidenced by recent literature. Therefore, we wish to re-emphasize the factors that have both helped and hindered our experimental process.

3.1 Boosting Generation via Better Understanding

The optimization objectives of Boogu are quality and efficiency, where quality is determined by the alignment between the user’s intent and the generated image, and efficiency is measured by the computational cost incurred to reach a satisfactory output. Our design is guided by several empirical observations accumulated during the development and deployment of large-scale text-to-image (T2I) systems:

- Many users issue very simple or underspecified instructions, such as a few keywords or a vague description, that nonetheless require non-trivial reasoning about context, common sense, or implicit visual conventions to be correctly interpreted and turned into a concrete generation target.
- Different tasks demand vastly different levels of generation capability: rendering a single isolated object is fundamentally easier than composing a multi-subject scene with precise spatial relations or stylistic constraints. Always invoking the strongest model is therefore computationally wasteful and offers diminishing returns on easy requests.
- Even the strongest Vision-Language Models (VLMs) suffer from inherent limitations on certain fine-grained capabilities, such as counting, precise spatial reasoning, and attribute binding, and relying on a single monolithic model to handle every aspect of understanding tends to inherit these weaknesses wholesale. Such limitations are further amplified when the system prompt becomes excessively long, forcing the model to attend to a large number of instructions at once and to lose focus on the most relevant ones for a given input.

To address these challenges in a unified manner, Boogu places understanding at the center of its design and decomposes it along three complementary dimensions. Each targets a distinct source of error in current text-to-image pipelines, and together they form the backbone of our quality–efficiency trade-off:

- **Understanding user intent.** The first bottleneck of generation quality lies in interpreting what the user truly wants, which spans two complementary regimes. (1) When a prompt is short yet precise, its intended semantics must be understood and faithfully preserved rather than over-rewritten or diluted, which calls for a sufficiently strong text encoder that can accurately capture the intended meaning. (2) When a prompt is vague or verbose, a stronger agentic prompt rewriter, built on better rewriting principles and a more capable VLM backbone, is needed to translate it into a faithful, generator-ready form while reasoning about the underspecified details. Because both the text encoder and the rewriter operate upstream of the generator, improvements here propagate to every downstream module.
- **Understanding the training images.** On the data side, the captioning strategy fundamentally determines what the model can and cannot learn: a generator can only acquire the concepts, attributes, and compositional patterns that its captions actually describe. Better image–caption alignment covering not only salient objects but also style, spatial layout, counts, and fine-grained attributes is therefore essential to inject the desired capabilities during training, and to prevent systematic blind spots from being baked into the model weights.
- **Understanding the task complexity.** Finally, on the inference side, Boogu explicitly estimates the difficulty of each request before deciding how to serve it. This complexity-aware routing directs simple requests to lightweight models and reserves the strongest generators for genuinely hard cases requiring deep reasoning or high-fidelity composition, yielding a principled trade-off that improves average quality while keeping the compute budget under control.

3.1.1 Instruction Encoder: The Sensor of Text-to-Image Models

We observe that many strong open-source text-to-image models are limited not by their generative components, but by an inaccurate understanding of the text description. Every text-to-image model relies on an “instruction encoder”, albeit in different forms, and we regard this encoder as analogous to a sensor: it defines the upper bound on the information available to the rest of the system. A weak encoder may discard important information before it interacts with image tokens, and because the instruction encoder is typically frozen during training, this limitation persists across both training and inference. In such cases, additional inference-time computation offers limited benefit, since no downstream process can recover information that was lost at the input. We therefore argue that a stronger encoder provides more accurate text understanding and higher-quality representations, and consequently yields better text-to-image generation. This view is consistent with several prior works that either report a similar pattern or adopt a stronger VLM as a default design choice, including Qwen-Image 2.0 [130] and Ideogram 4.0 [2].

Table 7 reports GenEval [42] scores as a function of encoder scale. Holding the training data, hyperparameters, and DiT architecture fixed—aside from minor adjustments required for tensor-dimension alignment—we find that *scaling the frozen instruction encoder alone yields consistent improvements in generation quality*. This indicates that the encoder’s capacity to comprehend and represent instructions is a fundamental determinant of text-to-image performance: a more capable encoder translates directly into better generation. We emphasize that these gains do not come from the parameter count itself, but from the encoder’s underlying ability to understand and encode instructions.

Owing to resource constraints, we did not evaluate larger encoders; however, within the range we examined, we observed no saturation in performance as the encoder scaled. While a stronger encoder is expected to further improve results, we also prioritize accessibility for open-source users, and therefore adopt Qwen3-VL-8B [3] as the instruction encoder—a choice that balances generation quality against deployment cost.



Figure 16 A single character changes everything. The two prompts express the same intent—a dragon and a phoenix in paper-cut style—and differ only by the conjunction “and”: “the dragon phoenix brings prosperity” (Prompt 1) versus “the dragon and phoenix bring prosperity” (Prompt 2). This single extra character breaks the fixed four-character idiom that the model has learned as a unit, producing a markedly different image despite the unchanged meaning.

Instruction Encoder[135]	Qwen3-1.7B	Qwen3-4B	Qwen3-14B
GenEval Score $\uparrow$	0.6034	0.6251	0.6477

Table 7 Effect of instruction-encoder scale on GenEval. We vary only the instruction encoder [135] while holding the DiT fixed at 1B parameters (with all other settings unchanged). GenEval scores improve monotonically as the encoder scales from 1.7B to 14B, with no saturation observed. These are intermediate results from models that are not yet fully trained, reported to show the relative trend across encoder scales rather than final performance.

3.1.2 Prompt Rewriter: A Translator, Not an Enhancer

In real-world applications, user prompts are frequently ambiguous, occasionally unsafe, and often require reasoning to be correctly interpreted—precisely the challenges that photographers, designers, and painters confront in their daily work. Without a strong translator that aligns user intent with the generated image, users may remain dissatisfied with the result even when the image exhibits high-quality detail and aesthetic appeal. A capable prompt rewriter is therefore essential to bridge the gap between underspecified user input and the structured, unambiguous conditioning that the generation model expects.

Employing a prompt rewriter has become a common practice among open-source models [19, 81, 130]. However, existing approaches exhibit several limitations. First, many rewriters produce excessively long outputs, introducing redundant or hallucinated details that drift from the user’s original intent and inflate inference cost. Second, they typically operate in a single forward pass, without the capacity to reason over ambiguous or compositional instructions before committing to a rewrite.

Motivated by these limitations, we adopt an agentic prompt rewriter that treats rewriting as a reasoning process rather than a single-shot transformation. Rather than expanding a prompt indiscriminately, our rewriter interprets user intent, resolves ambiguity, and produces a faithful, appropriately scoped instruction for the generation model—acting as a translator of intent rather than an enhancer of content. Notably, recently several concurrent work also proposes agentic prompt rewriter [60, 81].

Figure 17 shows an example that the prompt rewriter significantly boost overall performance. We utilize a reasoning-capable prompt rewriter to address several fundamental difficulties in text-to-image generation that the generation model alone struggles to resolve, as shown in Figure 18b.

- **Reasoning.** Prompts that specify a target indirectly—e.g., “the animal that lays eggs and produces milk” or “draw the result of 7×8 ”—require multi-step inference before a concrete visual subject can be determined. The rewriter resolves such implicit descriptions into explicit, renderable instructions.
- **Counting.** Generation models are notoriously unreliable at producing a precise number of objects. By explicitly enumerating instances in the rewritten prompt (e.g., expanding “five apples” into a structured, position-aware description), the rewriter alleviates miscounting and improves numerical fidelity.
- **NSFW prompts.** Unsafe or policy-violating requests can be identified at the rewriting stage and sanitized before reaching the generation model, providing an interpretable first line of content moderation.
- **Text rendering.** The rewriter brings substantial gains here, as user intents for adding text are frequently ambiguous or imprecisely formatted (e.g., “add a title” without specifying the exact wording or style). By resolving these into explicit, well-structured specifications, fixing the precise string and placement, the rewriter markedly improves both the accuracy and the visual quality of the rendered text.

While prompt rewriting is already widely adopted to improve generation quality, we re-emphasize a property that is often overlooked: a well-designed rewriter should be lower-bounded by the identity transformation. That is, when a user’s prompt is already clear, well-specified, and free of conflicts, the rewriter should leave it essentially unchanged rather than embellish it. Under this design principle, rewriting can only help and never hurt. In the worst case it preserves the original intent verbatim, and in the better cases it resolves ambiguity, underspecification, or conflict. This stands in contrast to rewriters that compulsively expand every prompt, which risk injecting hallucinated details and drifting away from a user request that needed no modification in the first place. Guaranteeing this lower bound is precisely what distinguishes a translator from an enhancer: the former intervenes only when intent is unclear, whereas the latter alters the input unconditionally.

Stronger vision-language models amplify the efficacy of prompt rewriting. As shown in Figure 18a, we observe a clear positive correlation between a model’s intrinsic world knowledge and reasoning capacity and the quality gain it delivers through rewriting. Given an identical user query, a more capable backbone draws on richer world knowledge and stronger compositional reasoning to produce more coherent, intent-faithful, and context-aware rewrites. This gap is, in fact, one concrete manifestation of the broader divide between open- and closed-source models—the rewriting ability is bounded not by the rewriting pipeline itself, but by the knowledge and reasoning embedded in the underlying model.

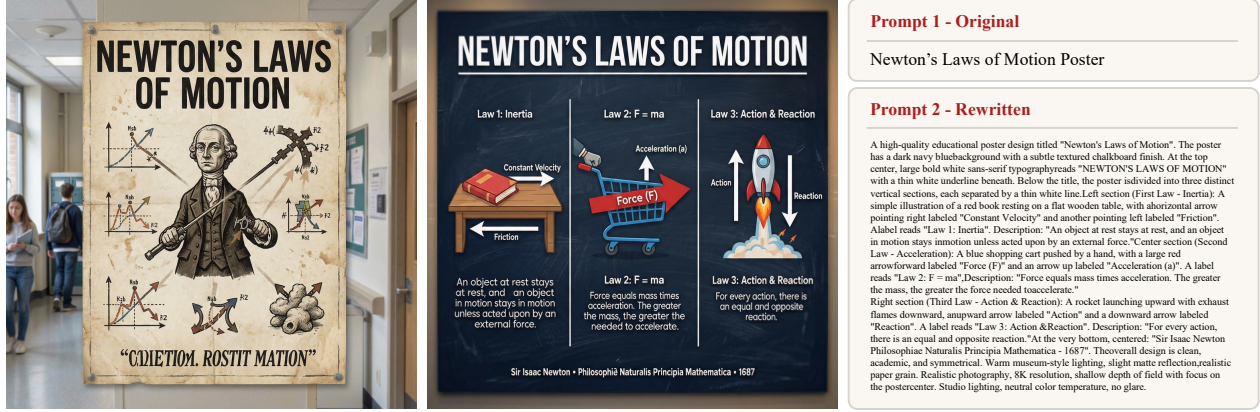


Figure 17 Effect of automatic prompt rewriting on text-to-image generation. The left two columns show the images produced from the original prompt (left) and its rewritten version (center); the right column lists the two prompts. Prompt 1 is a terse user request, “Newton’s Laws of Motion Poster”, which leaves layout, style, color, content, and typography entirely unspecified. Prompt 2 is the output of a prompt rewriter that expands the same intent into a detailed, structured specification: an explicit title and underline, a dark navy chalkboard background, three vertical sections with named laws, concrete illustrations (a book, a shopping cart, a launching rocket), labeled force and motion arrows, per-section descriptions, an attribution footer, and photographic rendering directives. The rewritten prompt removes ambiguity that the model would otherwise resolve arbitrarily, yielding a far more coherent, on-topic, and publication-quality poster. This illustrates that prompt rewriting acts as a bridge between a user’s high-level intent and the dense, descriptive input on which diffusion models [97] perform best.

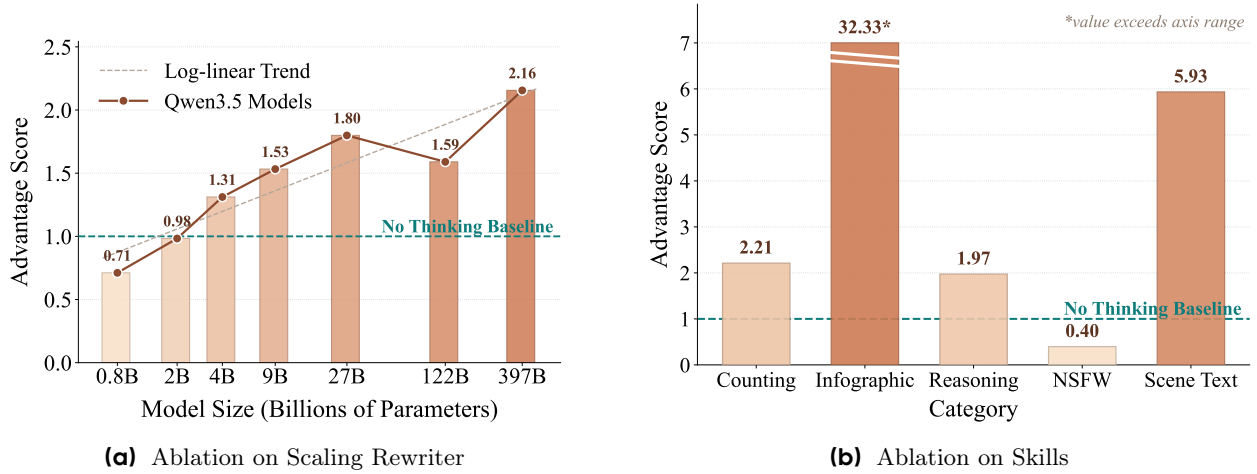


Figure 18 Both scaling up the thinking rewriter and introducing capability-specific PE skills effectively improve prompt rewriting quality. (a) Effect of different thinking models on rewriting. Compared with the no-thinking baseline, larger thinking models generally achieve higher advantage scores, indicating that stronger world knowledge and reasoning improve prompt rewriting quality. Among them, 122B and 397B are MoE models; the drop of 122B relative to 27B may be attributed to the MoE architecture or routing behavior, yet the overall trend remains that stronger models yield larger rewriting gains. (b) Effect of dimension-specific skills on the corresponding capabilities. After introducing dedicated skills for *Counting*, *Infographic*, *Reasoning*, *NSFW*, and *Scene-Text*, the advantage scores of the corresponding dimensions all change, with *Infographic* and *Scene-Text* showing the most notable gains, demonstrating that capability-oriented skill design effectively enhances the prompt rewriter in target scenarios. The *NSFW* score is below 1 because unsafe prompts are intentionally sanitized, which may reduce literal prompt satisfaction while improving *NSFW* alignment. The advantage score is defined as $S_{adv} = (N_{win} + N_{tie}^{good} + N_{tie}^{bad}) / (N_{lose} + N_{tie}^{good} + N_{tie}^{bad})$, where $S_{adv} > 1$ indicates an advantage over the baseline.

We further observe that this latent capacity is highly contingent on the system prompt. A poorly structured system prompt acts as a bottleneck that suppresses the model’s potential, whereas a carefully calibrated one serves as a catalyst that unlocks it. Consequently, realizing the full benefit of a strong backbone requires co-designing the rewriting prompt alongside the model.

Different text-to-image models call for different prompt rewriters. This is because each generation model is trained under its own conditions. First, models inject human preference along different axes: some are tuned toward photorealism, others toward illustration or aesthetic stylization, so the same prompt elicits different default behaviors. Second, models adopt different captioning strategies during training, meaning they expect input phrased in different styles and structures. A rewriter tuned for one model’s caption distribution may therefore produce out-of-distribution input for another, degrading rather than improving its output.

A direct consequence is that comparing two models under an identical prompt is often unfair: the prompt may sit squarely within one model’s training distribution while falling outside the other’s. A meaningful comparison should instead pair each model with its own appropriately adapted prompt, evaluating each at its intended operating point rather than under a shared but model-agnostic input.

3.1.3 Captions: Good Supervision Requires Understanding User Demand

The captioning strategy for training images must be designed around anticipated user demand. Captions serve as the supervisory signal that defines what the model can learn: if a capability is never described in the captions, the model has little chance to acquire it, and injecting that ability at inference time becomes extremely difficult, as neither prompt rewriting nor additional sampling can recover information that was never present in the supervision. Improving the alignment between an image and its caption is therefore central to the final quality of a text-to-image generation model.

Building such a high-fidelity captioning pipeline, however, is non-trivial, and existing practices face two recurring problems:

- **Limited capability of moderate VLMs.** Mid-sized vision-language models [3, 4, 146] often produce inaccurate captions, miscounting objects, hallucinating attributes, or misjudging spatial relations, and this noisy supervision is faithfully inherited by the generation model.
- **Prompt sensitivity of strong VLMs.** Even a strong VLM [40, 40] yields markedly different captions under different captioning system prompts. Without careful calibration, this inconsistency propagates as conflicting supervisory signals across the training set.

To address these issues, we systematically benchmark a set of candidate VLMs, including both open-source models such as Qwen2.5-VL-7B, Qwen2.5-VL-32B [4], Qwen3-VL-8B [3], and InternVL3-8B [146], as well as closed-source models such as Gemini-2.5 [40] and Gemini-3 [46]. We measure their captioning accuracy along each aspect of interest, including object counting, spatial relations, attribute binding, style, and text rendering. Rather than relying on a single model or a single system prompt, we select, for each aspect, the VLM and prompt configuration that performs best, and aggregate their outputs into a unified, demand-driven caption. For dimensions that most VLMs fail to caption accurately, we further resort to expert models or human experts to annotate the corresponding tags.

This per-aspect design yields consistently more accurate captions than any single-VLM baseline, and the resulting improvement in image-caption alignment translates into measurable gains in the corresponding generation capabilities of the trained model.

3.1.4 Model Router: The Right Model for the Right Task

For a human painter, the time needed to complete a painting varies greatly from a simple sketch to an intricate composition. Motivated by this observation, we design our system in a similar spirit: an agent decides which model to invoke, thereby controlling the trade-off between quality and efficiency.

Such practices are already common in the open-source community, where experienced developers and artists select different models, adjust the number of diffusion steps, and so on. While it is unclear whether top-tier

systems adopt similar strategies, one telling fact is that GPT-Image-2 [94] can take up to $100\times$ longer than Z-Image-Turbo [138] to generate a very simple prompt such as “a beautiful girl”. The same phenomenon arises within our Boogu-Image-0.1 series: in many cases the outputs of Boogu-Image-0.1-Turbo and Boogu-Image-0.1-Base are nearly indistinguishable, yet their inference costs can differ by 50 to 100 times.

Building on this insight, Boogu adopts an analogous strategy: rather than committing to a single model, it uses an agent to route each request to the most suitable variant in the Boogu-Image-0.1 series according to the estimated difficulty of the prompt, reserving the heavier Base model for genuinely complex requirements while handling simpler ones with the fast Turbo model. Similar ideas have been explored in the LLM community, where routing frameworks such as FrugalGPT [25], RouteLLM [92], and Hybrid-LLM [31] dynamically dispatch queries to models of different capacities to trade off quality against cost.

Admittedly, employing multiple models introduces additional system complexity. Nevertheless, given the substantial cost savings, we believe this direction is well worth pursuing.

3.2 General Discussion on Boogu Experiments

In this section, we present discussions based on the experimental evaluation of the Boogu model series. While these observations may appear trivial at first glance, they are in fact of critical importance.

3.2.1 Discussions Beyond Leaderboards

Inference-time cost should be considered in evaluations. A unified multimodal understanding-and-generation system differs fundamentally from a standalone text-to-image model: both the understanding and the generation components contribute substantially to end-to-end performance, and neither can be evaluated in isolation. This coupling has direct implications for inference cost. Strong closed-source systems such as GPT-Image-2 typically incur higher latency than open-source counterparts, in part because their quality reflects additional computation that is rarely accounted for in standard comparisons. However, even the most popular benchmark, LMArena [1], still does not take inference-time cost into account.

As shown in Figure 19, a broad class of techniques can improve output quality at the expense of inference time, including best-of-N sampling, self-reflection, prompt rewriting, and model ensembling [142]. As a recent example, GEMS [53] combines agents with Z-Image to outperform Nano-Banana on several benchmarks. These methods trade compute for quality along a continuous curve, which means that any quality figure reported without an accompanying latency budget is incomplete: the same model can occupy many points on this curve depending on how much test-time computation it is permitted to use. A meaningful comparison must therefore fix the inference budget across systems, or report quality and latency jointly as a Pareto frontier rather than as a single scalar.

In practice, however, most models report quality metrics without accounting for inference cost. This omission systematically favors methods that consume more test-time computation and obscures the genuine efficiency trade-offs that determine viability in deployment. A model that achieves a marginal quality gain at several times the latency may be preferable on a leaderboard yet unacceptable in a production setting.

We acknowledge that fair latency comparison is itself difficult. Closed-source systems must accommodate additional production constraints—including NSFW filtering, political-sensitivity handling, bias mitigation, and broader safety checks—that contribute to their measured runtime but are orthogonal to core generation quality. Isolating the cost attributable to generation alone is therefore non-trivial, and any cross-system latency comparison should be interpreted with this confound in mind.

Despite these challenges, we maintain that inference time must be incorporated as a primary evaluation axis rather than a secondary consideration. Quality and latency are not separable objectives; they jointly determine the practical utility of a model. Evaluating either in isolation produces conclusions that fail to transfer to real-world deployment.

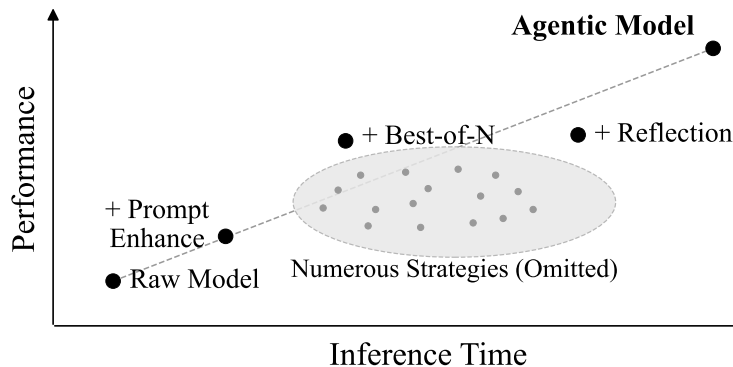


Figure 19 Trade-off between inference time and performance across strategies. Performance progressively improves from the Raw Model (bottom-left) to the Agentic Model (top-right) by integrating techniques (e.g., Prompt Enhancement, Reflection, Best-of-N), albeit at higher inference costs. The gray region denotes intermediate strategies omitted for clarity, and the dashed curve illustrates the overall performance-efficiency trend.

Over-reliance on a single baseline in A/B testing is methodologically flawed. During the intermediate phase of the project, we evaluated progress primarily through pairwise A/B comparisons against a single strong reference model. In retrospect, this framing was methodologically flawed. For open-source development, the appropriate evaluation target is not whether a model outperforms a specific competitor, but whether it improves upon the aggregate capability already available within the community. As shown in Figure 20, a favorable result against an individual baseline is both easy to obtain and prone to over-interpretation, since it reflects performance relative to a single point rather than to the prevailing capability frontier. The meaningful comparison is therefore A versus the union of all open-source work, that is, the effective capability envelope a user can assemble from currently available models and tools. We adopted this perspective only in the later phase of the project, after which our evaluation shifted from comparison against a fixed reference to comparison against this collective frontier.

This consideration further motivates designing evaluation sets around target application scenarios. A benchmark is informative only to the extent that it poses the questions that arise in real usage. We found that scenario-driven evaluation, using prompts derived from genuine production workflows, assessed by human raters, and aligned with deployment objectives, provides substantially more discriminative signal than static academic metrics. While such evaluation incurs higher construction and annotation cost, its correspondence to real-world conditions is what renders the resulting iteration reliable.

We note that strong closed-source models such as the Seedream [14, 15, 39] series have already adopted multi-dimensional comparison, yet this practice remains uncommon in the open-source community. While many open-source works attempt to approximate it by combining several benchmarks, this fallback is undermined by the flaws in public benchmarks discussed above.

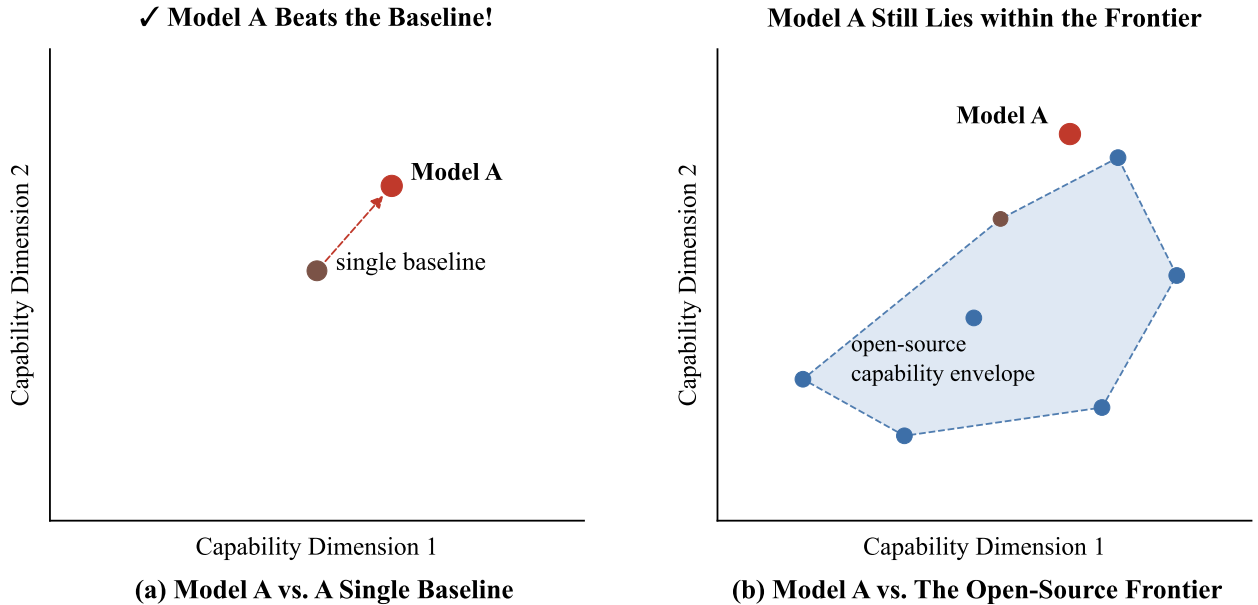


Figure 20 Single-baseline comparison versus the open-source capability frontier. (a) Compared against a single reference model, Model A appears superior on every dimension, inviting an over-optimistic conclusion. (b) When all available open-source models are considered jointly—forming a capability envelope (dashed region) that users can draw on in practice—Model A still lies largely within this envelope. The meaningful target for open-source development is therefore to advance the collective envelope, not to outperform any single baseline.

Open challenges for future evaluation. Even after reframing our evaluation methodology, several problems remain unresolved. Below, we highlight a few representative challenges to serve as a reference for the community, though this list is by no means exhaustive.

- **ImageNet-like T2I benchmarks are required.** The text-to-image community still lacks a standardized benchmark with the rigor that ImageNet [30] brought to recognition [52, 69, 105]. In particular, it lacks a clear and enforced separation between training and test data. At a minimum, models should control their training and test splits so that evaluation prompts (and the images associated with them) are guaranteed to be unseen during training. In current practice this discipline is largely absent: web-scale training corpora are scraped indiscriminately [61, 102, 111, 112], and popular evaluation prompts or their near-duplicates frequently leak into the training set. As a result, reported gains may reflect memorization rather than genuine generative capability [20, 33, 116], and cross-model comparisons become unreliable. We argue that establishing a benchmark with documented data provenance, held-out test prompts, and explicit leakage controls is a prerequisite for measuring real progress in text-to-image generation.
- **World-knowledge evaluation.** It remains unclear how to systematically assess whether a model captures real-world physical laws, spatial relationships, and causal structure [16, 49, 78]. Existing benchmarks [42, 59, 91] provide little coverage of these properties, and the failures they miss are often subtle. For example, a model may render a glass of water with physically inconsistent refraction, place a shadow on the wrong side of a light source, depict a mirror reflection that does not correspond to the scene, or generate a clock whose hands are inconsistent with a specified time. More demanding cases require compositional and causal reasoning. For instance [18, 78, 140], “a half-filled glass that has just been knocked over” requires the model to represent both the resulting spill and the residual liquid. Such errors are easy for humans to identify but difficult to capture with any automated, scalable metric [54, 58, 134], and we do not yet have a reliable protocol for measuring them.
- **Ambiguous semantics.** User prompts are frequently underspecified, and the space of acceptable outputs is correspondingly large [68, 110, 137]. An instruction such as “a nice living room” admits many valid interpretations, conditioned on user intent regarding style, lighting, and furnishing. The difficulty is compounded when prompts contain culturally dependent or contextual terms: “a traditional breakfast” implies different scenes in different regions, and “a formal outfit” varies with occasion and culture. Because no single reference output is correct, standard reference-based metrics are ill-suited to this setting [58, 71, 96, 106], and evaluating whether a model resolves ambiguity in a manner consistent with human expectations, rather than collapsing to a generic or averaged output, remains an open problem.
- **Inference time.** Quality and latency are in direct tension [56, 85, 90, 117]: techniques that improve fidelity, such as best-of- N sampling [134], prompt rewriting [51], and model ensembling [98], do so at the cost of additional computation. A model that achieves a marginal quality gain at several times the latency may rank favorably on a leaderboard yet prove impractical in interactive applications, where response time is itself a component of user experience. Identifying the appropriate quality and latency Pareto frontier for each deployment scenario, whether real-time interaction, batch generation, or offline production, remains an unsolved problem.
- **Dynamic benchmarks.** These observations motivate the development of evaluation protocols that are dynamic, contamination-free, and aligned with human judgment [5, 27, 76]. The underlying risk is a classic instance of Goodhart’s Law [28]: once a benchmark becomes an optimization target, it ceases to be a reliable measure. Engineering efforts are consequently driven toward over-optimization of specific benchmark formats, which can inadvertently degrade a model’s instruction-following and conversational capabilities in deployment. As a result, a model that surpasses its predecessor by 10% on academic benchmarks may nonetheless exhibit inferior performance in complex, open-ended user scenarios. In the absence of such protocols, public benchmark scores should be interpreted with caution, and evaluation should remain anchored to the criterion that matters: whether a model enables real users to solve tasks.

3.2.2 Data Ablation Evaluation: Just Enough Data Is Not Enough

How much data do we need? There are many open-source works that disclose their pretraining data. However, few of them discuss why a certain amount of data is required. The cost of training a foundation model, such as a text-to-image model, is highly correlated with the scale of the training data. Only a handful of organizations can afford to train such models, as the cost is prohibitive, including the computing resources, the disk storage needed to hold the data, and the expense of transferring it. Reducing this cost would open the problem to more organizations and broaden research in this area.

Prior excellent open-source models like Qwen-Image [130], Hunyuan-Image [19], Krea2 [72] mentioned that they utilized billions of images in the pretraining stages. Lens [22] utilizes fewer data but also require 800M images for training, which is close to 1 billion images. While there are some other interesting models that use fewer data but their performance is somehow not to close strong models mentioned above.

Boogu-Image-0.1 uses only 208.62 million unique images to achieve competitive performance, outperforming baselines that rely on substantially more training data and compute across many evaluation settings. In our opinion, the required volume of training data is mainly determined by two factors: the scope of knowledge, and the number of network parameters. As a foundation model, we usually hope it has good generalization ability and a wide range of knowledge. As the scope of knowledge grows, more training data is needed to represent it, consistent with studies of scaling laws [64, 77].

To reduce training cost, Boogu adopts two strategies aimed at lowering the volume of training data. First, we adopt a DiT with 10 billion parameters. Rather than scaling up model size, we deliberately keep the model compact so that it remains accessible to the open-source community, where many users and developers do not have access to high-end GPUs. This choice shifts the burden of efficiency onto data curation rather than raw model scale. Second, it is well known that internet-scale data exhibits a severe long-tail distribution: for a popular concept, the available data can be orders of magnitude larger than that of a rare one. Take landmarks as an example. As shown in Figure 21, data availability is heavily skewed toward a few iconic landmarks, with the Eiffel Tower attracting far more search interest than the rest. We argue that such an abundance of head data is unnecessary for training: beyond a certain point, additional examples of an already well-covered concept yield diminishing returns rather than better performance.

Note that there is a trade-off between the cost of constructing a balanced distribution and the cost it saves during training. Such careful data curation is only worthwhile when the available compute is limited; otherwise, constructing a balanced distribution may cost more than simply training on the unbalanced pretraining data.

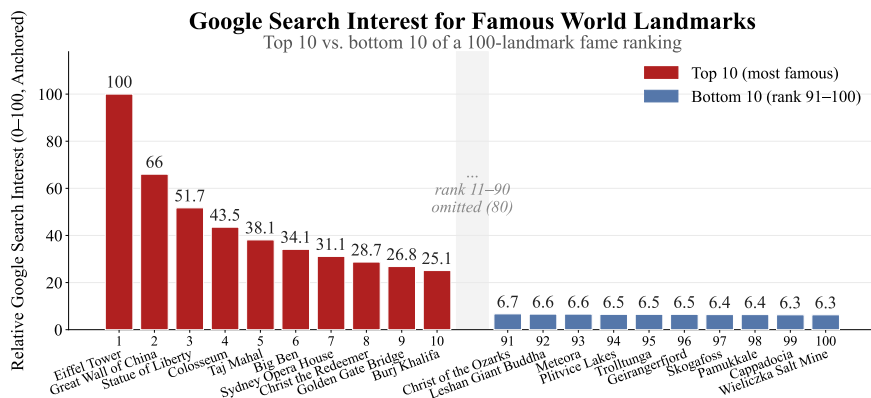


Figure 21 Google search interest across famous world landmarks. Each landmark is ranked by general global fame; the bars show the top 10 (most famous, left) and the bottom 10 (ranks 91–100, right), with the intervening 80 landmarks omitted for clarity, as marked by the central break. The vertical axis reports relative search interest on Google Trends (indexed to 0–100 and anchored to the top-ranked landmark so that all bars are comparable), not absolute query volume. Interest falls off steeply: the most famous landmarks attract many times the search interest of those in the long tail. *Note:* the fame ordering is editorial rather than derived from a formal metric, so the figure is intended to illustrate the overall distribution rather than to provide an exact measurement.

Explicitly captioning visual flaws instead of filtering flawed data. Very few categories of data actually need to be filtered out if your goal is to build a general-purpose image generation model. While training a foundational text-to-image model requires high-quality, aesthetically pleasing images with optimal exposure and composition to ensure generation fidelity, **suboptimal data, such as underexposed, poorly composed, blurry, or watermarked images, should not be entirely discarded.** Unlike recent data pipelines such as Qwen-Image [130], Z-Image [138], and JoyAI-Image [118], which commonly emphasize filtering images with visible artifacts or quality defects, we argue that these samples can remain valuable if paired with detailed and accurate captions. By explicitly describing all visual and logical content alongside their specific flaws (e.g., detailing the presence of watermarks or the exact nature of the blurriness), these images can be integrated into the training set at an appropriate ratio. This not only helps the model learn fundamental visual patterns but also enables it to understand negative visual elements, thereby acquiring a more comprehensive and controllable generation capability. In principle, almost any 2D image can be utilized; the real challenge is not whether to include such data but how to balance it, which demands extensive fine-grained tagging or manual annotation to enable precise control.

Figure 22 illustrates this idea with several common artifacts found in open-source images. Rather than discarding these samples, we retain them and pair each with a caption that explicitly names the defect: a visible watermark (top-left), color noise from a low-light night shot (top-right), motion blur caused by a slow shutter speed (bottom-left), and severe overexposure (bottom-right). By grounding each artifact in language, the model learns not only the underlying visual content but also the concept of the defect itself. This turns what would otherwise be filtered-out “bad” data into a signal for controllable generation, allowing the model to either avoid these artifacts or reproduce them on demand.



Figure 22 Common artifacts in open-source images, paired with captions that explicitly describe them. Simply filtering out such images would deprive the model of the ability to understand these artifacts; instead, we retain them with captions that name the defect, enabling controllable generation. **Top-left:** a watermarked image. **Top-right:** a low-light night photo showing color noise upon magnification. **Bottom-left:** motion blur from a slow shutter speed. **Bottom-right:** significant overexposure.

Quality over quantity: massive open-source datasets do not guarantee high performance. For the initial pretraining phase, we construct a foundational data mixture entirely from publicly available sources, comprising both single image-text pairs and interleaved text-image documents. Specifically, we collect representative single-image corpora, including COYO [13], DataComp [37], PixelProse [114], and BLIP-3o [23]. To further expand the overall data scale and enhance broad visual-concept coverage, we supplement these with a large-scale interleaved corpus derived from OmniCorpus [74].

Our empirical findings demonstrate that relying solely on open-source datasets (here 187M in total) imposes a strict ceiling on model performance, rendering them inadequate for training state-of-the-art models. Despite their massive volume, open-source corpora suffer from pervasive quality issues, including high semantic noise, factual errors, and severe data duplication.

More importantly, these datasets lack the descriptive depth required for high-fidelity generation; their captions are predominantly brief and shallow, failing to capture the fine-grained attributes, spatial relationships, and dense image-text alignments necessary for precise and controllable synthesis. Furthermore, open-source data is highly skewed toward generic web content, leaving a critical void in high-quality, domain-specific (e.g., scientific, logical, and localized) data. To bridge the gap to proprietary frontier models, developers must move beyond public datasets and focus on building sophisticated data-refinement pipelines, high-quality synthetic data generation, and meticulous expert-level curation.

As shown in Figure 23 and Table 8, a model trained on data structured by a carefully designed, human-prior-based syllabus significantly outperforms one trained on massive (187M) open-source data in terms of instruction comprehension and following, world knowledge, and photographic quality.

Training Data	Qwen-Image-Bench (CN+EN) $\uparrow$	LongText EN $\uparrow$	LongText ZH $\uparrow$
Open-source Data	48.45	0.828	0.869
+Boogu Syllabus	53.65	0.952	0.969

Table 8 The Boogu Syllabus substantially outperforms open-source data. Under identical model and training settings, training on the curated Boogu Syllabus improves performance across all three benchmarks over training on raw open-source datasets. Best scores are **bolded**.

Prompt 1: A masked cat (anthropomorphized) grips a handrail in a subway car, its eyes looking tired; an advertisement in the car reads “AI, keep going!” Requirements: correct perspective, the cat’s anthropomorphism is reasonable and cute, the advertisement text is legible, and the lighting is consistent.

Prompt 2: Amidst heavy snowfall, a swordsman in a red cloak stands alone on the city wall, the wind whipping his cloak into sharp lines. Requirements: contrasting warm and cool colors (cold snow + red cloak); low-angle shot; sharp edges; natural pose.



Figure 23 Training on the Boogu Syllabus yields markedly better generation quality than open-source data. The left two images are generated by the model trained on open-source datasets, while the right two are generated by the model trained on the Boogu Syllabus (prompts shown above). Under identical settings, the Syllabus-trained model produces more accurate text rendering, more coherent composition, and better prompt adherence.

Cultural bias ingrained in open-source data persists despite targeted SFT. We identify a distinct capability gap across cultural contexts, where the model acquires Western concepts (e.g., Caucasian features) more readily than Eastern ones. We attribute this discrepancy to the inherent Western-centric bias in large-scale open-source pre-training data, which is heavily dominated by Western cultural contexts—a systemic imbalance that remains difficult to quantify given the sheer volume of the data. Our experiments reveal that even incorporating a large-scale, high-quality Chinese dataset during the SFT stage cannot fully rectify this bias. Because foundational worldviews and cultural associations are established during the massive pre-training phase, downstream SFT only scratches the surface of behavioral alignment without fundamentally altering the model’s latent representation space. As a result, the model still defaults to Western-centric patterns in its generations (see Figure 24), suggesting that cultural diversity must be addressed during pre-training rather than treated as an afterthought in fine-tuning.

As shown in Figure 24, we present image samples generated by our fine-tuned model using semantically equivalent Chinese and English instructions. Notably, even when prompted with Chinese instructions, the model exhibits a markedly high probability of generating Western faces or defaulting to Western cultural elements, such as architectural designs, street layouts, and clothing styles. While varying the random seed introduces some visual diversity, the predominant emergence of this Western-centric aesthetic remains highly consistent. This phenomenon clearly demonstrates the cultural bias discussed above, underscoring the urgent need for the open-source community to collect and curate datasets with greater cultural diversity.

EN: A street photography shot of an elderly scavenger with a deeply weathered and wrinkled face in the center of the frame. The scene includes a trash can and a traffic light in the background. High photographic texture, classic street photography aesthetic, cinematic lighting, photorealistic.

ZH: 生成一张街拍的照片，要有垃圾桶，红绿灯。画面中心有一个拾荒的老人，满脸沧桑。照片要有摄影质感，街拍质感。



Figure 24 Semantically equivalent prompts in different languages produce culturally biased outputs. The left two images are generated from the English prompt and the right two from the Chinese prompt (shown above), despite the two prompts describing the same scene. The model renders the subject and setting with markedly different cultural characteristics depending on the input language, revealing a language-conditioned cultural bias.

Syllabus-guided data curation with human priors significantly boosts performance. Pure visual understanding is inherently a “dimensionality reduction” task involving information compression—mapping rich visual signals to lower-dimensional text. Regardless of how exhaustive a natural language description is, it inevitably loses some of the original visual information. Consequently, generating an image or video based solely on even the most detailed text will likely yield results that differ from the original source. Conversely, Text-to-Image (T2I) generation is a “dimensionality expansion” task, granting the model immense creative freedom to synthesize countless valid visual variations that satisfy a given user prompt.

Recent studies on visual understanding models reveal several critical insights: (1) Model capabilities exhibit a severe long-tail distribution with respect to their training data [95]. While models excel at common concepts (e.g., cats or dogs), they struggle with rare elements due to the inherent long-tail nature of mainstream datasets. (2) Multimodal models require exponential data exposure to grasp a concept [124]; specifically, zero-shot performance grows linearly only with exponentially increasing exposures to similar data. (3) Scaling up model parameters alone cannot compensate for unseen data types [26]. If a specific data category is absent during training, the model cannot align its visual and textual representations, rendering it incapable of understanding that concept, regardless of model size.

Given that these limitations apply to the dimensionality reduction task like visual understanding, we argue that the dimensionality expansion task of T2I generation imposes even stricter requirements on data and training. To circumvent the aforementioned issues, we propose the following data curation principles:

- *Systematic and Fine-Grained Deconstruction:* Data serves as the textbook for model learning. Just as physics textbooks deconstruct knowledge into distinct branches (e.g., thermodynamics, electrodynamics) based on human priors to facilitate learning, systematically deconstructing data into fine-grained, logically interconnected components forms a structured syllabus that enhances the AI’s understanding.
- *Comprehensive Coverage:* The dataset must be extensively comprehensive; ideally, no required capability should fall outside the planned scope. We must proactively curate data for every target capability rather than relying on the model to achieve sudden “epiphanies” or zero-shot generalization through sheer scale. Simply put, if the model has never seen a cat during training, we assume it cannot generate one.
- *Ample Data per Category:* Each fine-grained domain must contain sufficient data volume. As empirically verified in Section 3.2.3, we estimate the requisite number of exposures needed for a model to memorize a concept and ensure that each data category meets this threshold during our curation process.

By adhering to these principles, we aim to mitigate the long-tail effect in model capabilities and systematically guarantee robust, grounded understanding, moving beyond the naive reliance on scaling data volume or model parameters in hopes of spontaneous emergence.

In short, mere data abundance is far from sufficient. To truly enhance a model’s comprehension and learnability, data must be structured and organized guided by human priors—a process that inherently injects human knowledge into the learning pipeline. Next, we detail the data distribution of our Boogu syllabus.

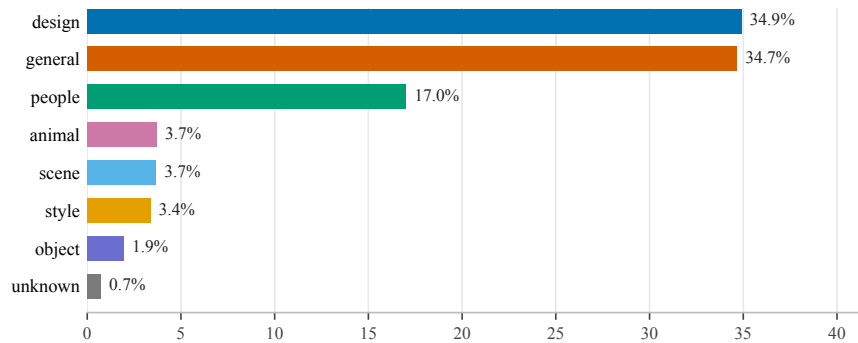


Figure 25 Macro-level data distribution of the Boogu syllabus.

A well-designed distribution can potentially reduce the required volume of training data. Guided by human

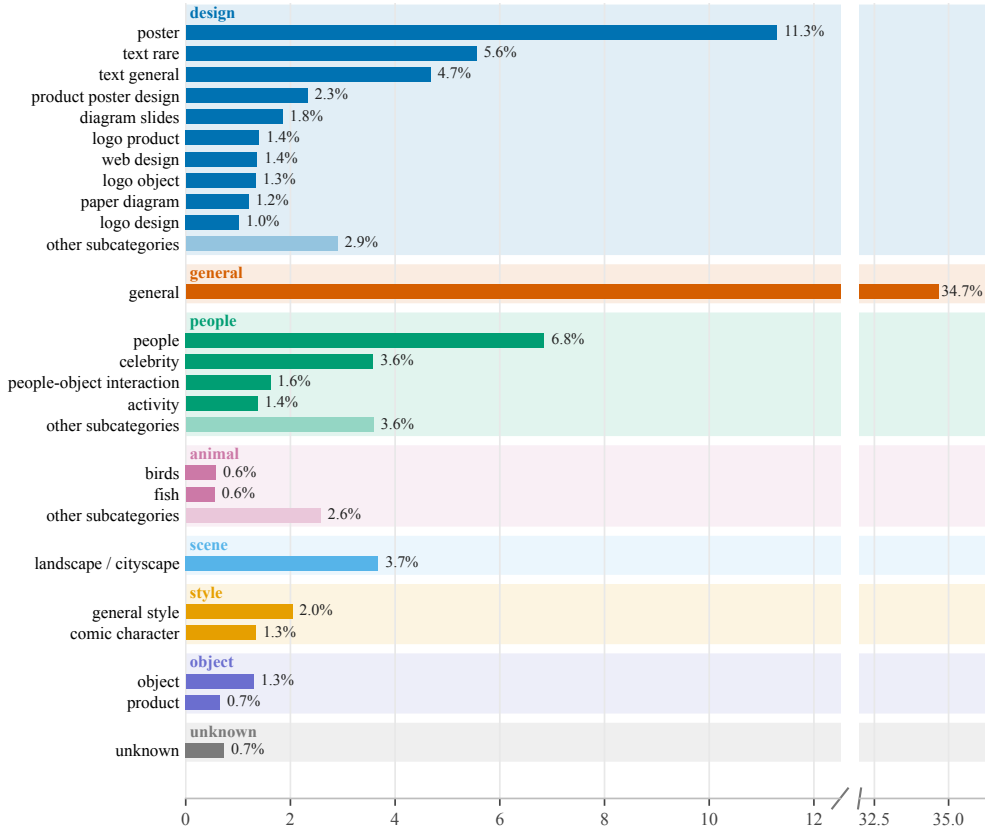


Figure 26 Micro-level data distribution of the Boogu syllabus.

priors, we deconstruct visual scenes and organize them by their underlying logical relationships, structuring the data into a syllabus that facilitates the model’s learning. Specifically, we assign higher sampling weights and construct fine-grained data for complex tasks. For example, as illustrated in Figures 25 and 26, graphic design constitutes a substantial and finely sub-categorized portion of our Boogu syllabus, reflecting its multifaceted nature: it requires interrelated capabilities—such as accurate typography, logo and poster design, and composition—that collectively determine the final visual quality.

These targeted strategies ensure that “**no required capability should fall outside the planned scope**,” enabling the model to systematically comprehend diverse tasks and accurately align with the necessary visual elements. In summary, our Boogu syllabus comprises 21.62M unique samples. During training, these are upsampled to 47.19M based on our predefined weighting ratios. Notably, our model’s comprehensive performance significantly surpasses that of models trained solely on 187M open-source samples (see Table 8).

Systematically enumerating vocabulary for reliable text rendering. To improve the model’s ability to render visual text, particularly Chinese characters, we propose a systematic data construction strategy based on vocabulary enumeration. This targeted approach proves far more effective than undirected data augmentation, which often suffers from coverage omissions and redundant sampling.

Our empirical analysis reveals that the model is virtually incapable of rendering unseen Chinese characters, requiring a minimum of 300 training exposures per glyph for accurate generation. To validate this, we constructed a subset of Chinese characters with varying frequencies (Figure 27). Models trained on this subset consistently and correctly generate characters observed over 300 times. Furthermore, for characters the model initially fails to render, introducing them for more than 300 exposures during subsequent training ensures stable and accurate generation (Figure 28).

Under resource and time constraints where enumerating the entire Chinese vocabulary is infeasible, prioritizing the 3,500 most frequently used modern Chinese characters [148] serves as a practical alternative. By ensuring each character appears at least 300 times in the dataset, the resulting model can satisfy everyday Chinese text rendering demands. Table 9 presents the performance gains resulting from optimization targeted exclusively at the 3,500 frequently used Chinese characters.

Ultimately, this proactive data construction strategy is significantly more efficient than blindly accumulating open-source data in an attempt to acquire text rendering capabilities through sheer volume.

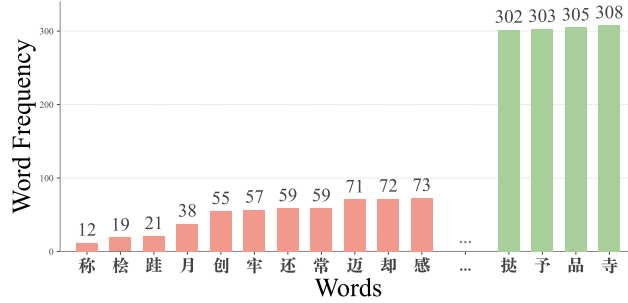


Figure 27 Per-character word frequency of a subset. The horizontal axis lists Chinese characters in the ablation subset, sorted by how often they appear in training (high to low), and the vertical axis reports each character’s generation accuracy. Characters seen at least ~ 300 times are generated reliably, whereas rarer characters degrade sharply—indicating that roughly 300 exposures suffice for accurate generation.



Figure 28 Correcting rare character rendering via targeted exposure scaling. The figure contrasts the rendering performance before (left two panels) and after (right two panels) optimization, i.e., after SFT on a curated synthetic dataset that scales the training exposure of each Chinese character beyond 300 times, the previously unrenderable character 抛 is now correctly and reliably rendered.

SFT Optim.	LongTextBench (ZH) $\uparrow$
Before	0.9055
After	0.9538

Table 9 Target-exposure scaling via SFT optimization substantially boosts the model’s performance on the Chinese subset of LongTextBench.

3.2.3 Ablation Experiments on Multimodal Learning and Understanding

Robustly memorizing an unseen concept requires crossing a quantifiable exposure threshold. To evaluate the model’s ability to memorize specific human identities, we incorporated 170 daily photographs of a volunteer (Guoxuan Chen) into an ablation dataset. Each image was annotated with four caption variations (Chinese/English, short/long descriptions). Through oversampling, we expanded this subset to 11,628 images (5,814 per language) and mixed it into a comprehensive 12.6M SFT dataset, constituting $\sim 0.13\%$ of the total data. The dataset was randomly shuffled to ensure uniform distribution. We trained the model using 128 devices with a global batch size of 1024, applying a constant learning rate of $1e-4$ after a 500-step warmup. Using a volunteer’s photos rather than celebrities ensures that the model had strictly no prior exposure to this identity during pre-training. The rightmost images in Figure 29 show the ground truth photos included in the training set: a long shot of the volunteer at $\sim 70\text{kg}$ (top right), a close-up at $\sim 75\text{kg}$ (bottom right, first), and a medium shot at $\sim 80\text{kg}$ (bottom right, second). Notably, the subject’s appearance varies slightly across these periods due to fluctuations in weight and personal condition. Note that the flow matching [79] model tends to learn an averaged representation of these variations.

The generated results from different training checkpoints are shown in Figure 29. The first four images in the top row correspond to 1k, 2k, 3k, and 5k steps, while the first three in the bottom row correspond to 6k, 8k, and 9k steps. We tested the model using Chinese prompts. At 1k steps, the model had encountered the volunteer’s photos approximately 1,000 times (including around 500 times with Chinese captions). As shown, the model begins to generate recognizable facial features around 5k steps and achieves full recognizability by 8k to 9k steps. This indicates that a model requires about 4,500 language-matched exposures during training to robustly memorize a completely unseen human identity based on its associated textual identifier.

Prompt: 这张横向构图的室内写实摄影照片主体是男青年陈国轩，他正端着一个生日蛋糕。画面采用近景拍摄，光线聚焦在陈国轩和蛋糕上。他拥有整齐的黑发，戴着银色金属细框眼镜，表情平和且带着一丝微弱的笑意。其上身穿着一件带有帆船线条图的白色T恤，外搭一件蓝白相间、带有大面积格纹和晕染质感的翻领长袖衬衫。陈国轩双手托着一个白色方形盘，盘中是一个圆形生日蛋糕，蛋糕侧边和底部洒满了淡黄色的杏仁碎，顶部点缀着新鲜的草莓半块、绿色的猕猴桃片和其他水果。蛋糕正中心插着一个圆形的金色透明牌，上面印有艺术字体的“Happy Birthday”，两根正在燃烧的橘红色细长蜡烛立在插牌侧前方，火光在金色背景处形成了明亮的亮斑。背景是虚化的浅色室内墙面，整体氛围温馨。画面左右两侧存在轻微的视角透视拉伸感，文字清晰可见。



Figure 29 The generated author identity progressively converges to the ground-truth subject. Top row: checkpoints at 1k, 2k, 3k, and 5k steps; bottom row: 6k, 8k, and 9k steps. Every 1k steps corresponds to approximately 1,000 exposures to the ground-truth data (500 with English and 500 with Chinese captions). The top-right ($\sim 70\text{ kg}$) and two bottom-left ($\sim 75\text{ kg}$ and $\sim 80\text{ kg}$) images are ground-truth training samples of the author at different weights. With increasing training, the model captures the subject’s identity with growing fidelity, using the same prompt.

Use reinforcement learning with care when training text-to-image models. Distributional diversity is a core requirement for a general-purpose text-to-image model: users expect to generate not only mainstream-looking subjects but also those that deviate from prevailing aesthetic norms. Preserving the breadth of the output distribution is therefore essential to broad usability.

A common technique is to apply reinforcement learning (RL) [82] to improve aesthetics and generation stability. However, heavy RL tends to collapse the output distribution toward a narrow domain. For instance, Seedream models [113] achieve impressive portrait results—the generated people are consistently beautiful and young, matching mainstream preferences—yet generating subjects that depart from this aesthetic becomes considerably harder and sometimes infeasible. The very alignment that raises average quality also narrows the reachable distribution. The very alignment that raises average quality also narrows the reachable distribution. As shown in Figure 30, given the prompt “A sixty-year-old Chinese woman”, models trained with heavy aesthetic RL tend to render the subject with polished makeup and idealized, glamorized features, whereas our model produces a more natural and realistic result. We emphasize that such RL-driven distribution shaping is not a weakness; it is a legitimate product strategy. It simply conflicts with our goal of maximizing diversity.

Guided by this objective, we deliberately avoid heavy aesthetic RL. Instead, we embed human preference within the understanding system rather than baking it irreversibly into the generator, allowing style—including attributes such as human faces—to be controlled flexibly at inference time while the generator’s underlying diversity is preserved. As a result, our model readily generates subjects across a wide aesthetic range, including those outside mainstream norms.

This does not mean abandoning RL entirely. RL remains valuable for targeted issues that do not compromise diversity, and it delivers real gains in reducing anatomical artifacts and improving text rendering. However, RL training incurs substantially higher computational cost: it requires reward-model inference, repeated sampling, and many additional optimization steps on top of the base training pipeline. To build Boogu in a cost-efficient manner, we therefore limit RL to the few targeted issues where it is indispensable—such as anatomical distortion and text rendering—rather than applying it broadly for aesthetic tuning.

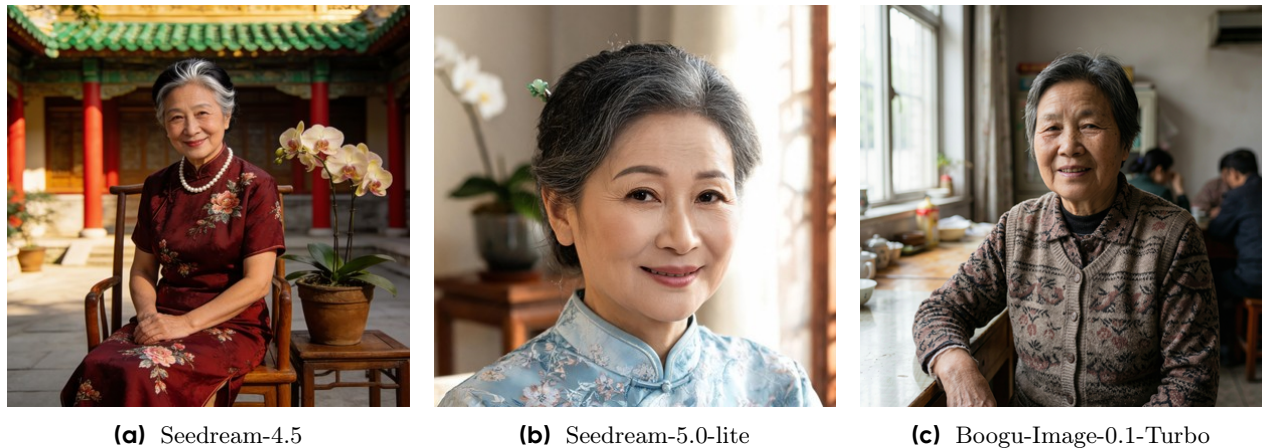


Figure 30 Generated images for the prompt “A sixty-year-old Chinese woman”. Models trained with heavy aesthetic RL (a, b) collapse toward mainstream preferences, rendering the subject with idealized, made-up features rather than the plain, natural appearance implied by the prompt, whereas our model (c) produces a more realistic result. We emphasize that such RL-driven distribution shaping is not inherently a weakness; it is a legitimate product strategy. It simply conflicts with our goal of maximizing diversity.

Dynamic time shifting does not scale trivially to 2K resolution. In diffusion and flow matching models, logit-normal timestep sampling combined with resolution-dependent timestep shifting (dynamic time shifting) has become a widely adopted standard [7, 36, 79, 84]. These strategies are renowned for optimizing the signal-to-noise ratio and accelerating convergence, particularly when training high-resolution models.

However, **during the development of the Boogu series---one of the pioneering open-source models natively trained at 2K resolution---we observed a phenomenon. Applying the default logit-normal and dynamic time shifting strategies significantly hinders the convergence rate at this high-resolution scale.** Consequently, models in the early to intermediate stages of training exhibit noticeable noise artifacts in their generated outputs. Here, we provide a detailed analysis. For standard logit-normal sampling, we have:

$$t = \text{Sigmoid}(u) = \frac{1}{1 + e^{-u}}; \quad u \sim \mathcal{N}(0, 1)$$

To model resolution-dependent timestep shifting, we typically employ linear interpolation to quantify how the number of latent tokens, n_T , influences the distribution of the timestep t . Calibrated by two pre-defined reference points (x_1, y_1) and (x_2, y_2) , the shift factor $\mu(n_T)$ is formulated as:

$$\mu(n_T) = k \cdot n_T + b; \quad k = \frac{y_2 - y_1}{x_2 - x_1}, \quad b = y_1 - k \cdot x_1$$

where n_T denotes the resolution-dependent token count in the latent space. The shifted timestep, t_{shift} , is then computed as follows (note that in our Boogu series, $t \rightarrow 0$ denotes the high-noise regime):

$$t_{\text{shift}} = \tau(t; \mu(n_T), \sigma) = \frac{t^\sigma}{t^\sigma + e^{\mu(n_T)} \cdot (1 - t)^\sigma}; \quad \sigma > 0$$

Fundamentally, this operation is equivalent to a translation in the logit space:

$$\text{logit}(t_{\text{shift}}) = \log\left(\frac{t_{\text{shift}}}{1 - t_{\text{shift}}}\right) = \sigma \cdot \text{logit}(t) - \mu(n_T); \quad \text{logit}(p) = \log \frac{p}{1 - p}$$

This reveals that the complex time-shifting reduces, in logit (log-odds) coordinates, to a simple affine transformation: a scaling by σ followed by a translation of $-\mu(n_T)$. With the common choice $\sigma = 1$, it degenerates into a pure translation by $-\mu(n_T)$. Consequently, a larger $\mu(n_T)$ —induced by higher-resolution images—shifts the overall distribution towards smaller t . While this aligns with the intuition that higher resolutions demand more computation at the high-noise stages to resolve global structures, applying this standard dynamic shifting strategy at 2K resolution leads to an *over-squeezing* effect (see Figure 32, right). This excessive shift overly degrades the signal-to-noise ratio during training, ultimately slowing the convergence rate.

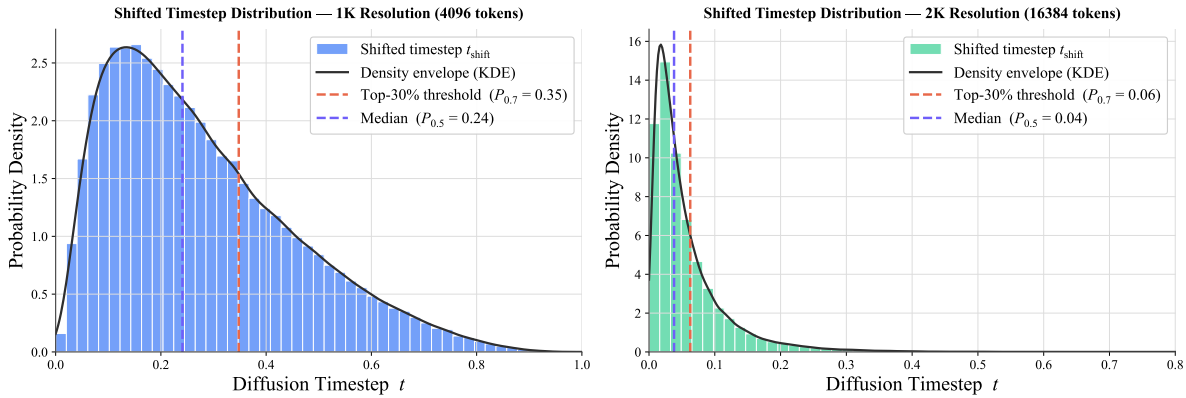


Figure 31 Shifted timestep distributions at 1K and 2K resolutions ($\sigma = 1$), calibrated using $(x_1, y_1) = (256, 0.5)$ and $(x_2, y_2) = (4096, 1.15)$. The right panel illustrates the over-squeezing effect observed at the 2K resolution.

Rectified Dynamic Time Shifting. To mitigate the over-squeezing effect discussed above, the dynamic time-shifting scheme must be rectified. The most straightforward remedy is to manually retune the reference points (x_1, y_1) and (x_2, y_2) so as to flatten the slope of the linear interpolation, thereby slowing the growth of the shift magnitude $\mu(n_T)$ as the token count n_T increases quadratically with resolution. However, directly flattening the slope leaves the shift insufficient for the low-resolution images (e.g., 512×512) that co-exist in mixed-resolution training. An alternative is to replace the linear map with a non-linear $\tilde{\mu}(\cdot)$, which in turn poses a non-trivial optimization problem. For instance, some works [131] adopt a logarithmic curve:

$$\tilde{\mu}(n_T) = \alpha \cdot \log(n_T) + \beta,$$

where one may take $\alpha = \frac{1}{2}$ and $\beta = -\log 20$; the factor $\alpha = \frac{1}{2}$ counteracts the quadratic growth of n_T with respect to the image side length (i.e., $\sqrt{n_T}$). Nevertheless, fitting (α, β) is itself an optimization problem, and the resulting monotonic curve may still mismatch the optimal shift across resolutions. In short, choosing an optimal $\mu(n_T)$ is a complex problem to which we do not claim an optimal solution; we leave it to the research community and to future work.

Instead, we adopt a simple yet effective scheme that clamps the effective token count:

$$\tilde{\mu}(n_T) = \mu(\min(n_T, n_{\text{cap}})) = \min(\mu(n_T), \mu(n_{\text{cap}})),$$

where, e.g., $n_{\text{cap}} = 4096$. This means that beyond the 1K-resolution regime, no additional time shifting is applied. To facilitate the analysis, we characterize the distribution of t_{shift} through its quantile function. Letting $P_q := Q_{t_{\text{shift}}}(q; \mu(n_T), \sigma)$ denote the q -quantile (i.e., the value P_q such that $\Pr[t_{\text{shift}} \leq P_q] = q$), we have:

$$P_q = Q_{t_{\text{shift}}}(q; \mu(n_T), \sigma) = \text{Sigmoid}(\sigma \cdot \Phi^{-1}(q) - \mu(n_T)); \quad q \in (0, 1), \sigma > 0, \Phi(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-s^2/2} ds,$$

where we set $\sigma = 1$ by default. Based on this expression we plot Figure 32. The family of curves $\{P_q\}$, whose rate and trend are governed by $\mu(n_T)$, could in principle be studied via mathematical analysis, which would be rather involved. Our piecewise construction sidesteps this complexity: **by substituting $\tilde{\mu}(n_T)$ for $\mu(n_T)$, we directly inherit the tuning effort validated at 1K resolution, seamlessly extend it to 2K training, and eliminate the over-squeezing artifacts**, namely slow convergence and noisy generations at early stage.

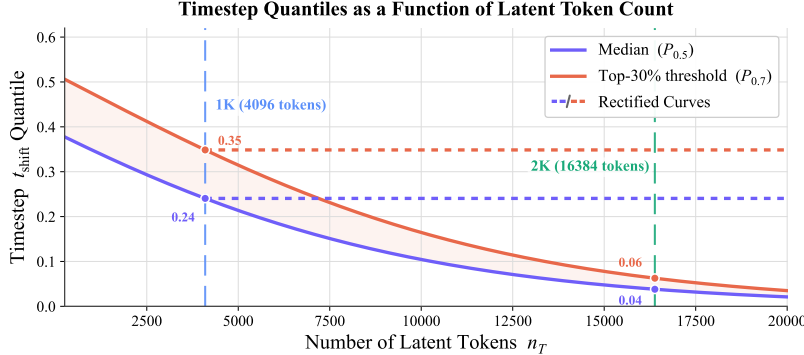


Figure 32 Timestep quantiles versus latent token count under dynamic time shifting. For a given token count n_T , we plot two quantiles of the shifted timestep t_{shift} , computed from $P_q = \text{Sigmoid}(\sigma \Phi^{-1}(q) - \mu(n_T))$ with $\sigma = 1$, $(x_1, y_1) = (256, 0.5)$ and $(x_2, y_2) = (4096, 1.15)$: the median ($P_{0.5}$, purple) and the top-30% threshold ($P_{0.7}$, orange). Solid curves use the original linear shift $\mu(n_T)$, which keeps growing with n_T and drives both quantiles toward 0 at high resolution (e.g. the median collapses from 0.24 at 1K to 0.04 at 2K), reflecting the over-squeezing effect. Dashed horizontal lines show our rectified scheme, which clamps the effective token count at $n_{\text{cap}} = 4096$ (the 1K regime) so that the percentiles remain fixed at their 1K values beyond this point. The rectification thus inherits the well-tuned 1K shifting behavior and extends it to 2K training without exacerbating over-squeezing.

2D structural awareness in DiT enriches photographic atmosphere. While standard Diffusion Transformers (DiTs) [97] flatten spatial representations into 1D sequences and thereby neglect the inherent 2D nature of images, we find that explicitly preserving the 2D matrix structure of the latent space can improve the photographic and cinematic atmosphere of the outputs, yielding richer textures.

For conventional DiT, at each flow-matching (or denoising) step, the model output is treated as a single flattened 1D update direction in a high-dimensional space, along which the current latent is advanced toward a clean-image hidden state [56, 79, 84, 86, 115, 117, 119]. Although convenient, this view discards a key inductive bias of the image modality: the DiT output is a tensor of shape $[B, C, H, W]$, whose $[H, W]$ dimensions form a meaningful grid aligned with the image plane rather than an arbitrary reshaping artifact. The mismatch is amplified by common normalization practices, which first collapse the 2D prediction into a 1D vector and then rescale its Euclidean length—altering only the magnitude while leaving the update direction and its spatial composition untouched. Even variants that additionally apply mean-centering remain blind to spatial organization, missing the opportunity to preserve informative variation and avoid overly smooth updates.

Inspired by recent optimizers that respect matrix/tensor structure instead of reducing updates to flat vectors [50, 63], we propose an alternative classifier-free guidance [55] scheme for DiT, termed *Boosted Orthogonal Guidance* (BOG), which applies a 2D normalization strategy directly to the matrix-structured prediction at each step. Figure 33 contrasts the generation results of BOG and conventional CFG. Notably, we believe that the matrix-structured view of DiT predictions opens up considerable room for further improvement, which we leave to future work and community exploration. A detailed discussion of BOG—including its motivation, usage and limitations—is provided in Appendix C.

Prompt 1: 画一只在森林中抓拍到的棕熊，棕熊旁边有一个路牌，写着“Watch Out!”。棕熊回头看这个路牌。图片要有真实感。

Prompt 2: 生成一张街拍的照片，要有垃圾桶，红绿灯。画面中心有一个拾荒的老人，满脸沧桑。照片要有摄影质感，街拍质感。

Prompt 3: 画一个篮球运动员在打篮球，胸前是8号球衣，球队是Boogu，场下有很多观众。这个照片是真实摄影风格。

Prompt 4: 画一个中国男人，手里举一个牌子，上面写着“好好学习!”。要真实照片的风格。



Figure 33 BOG produces more realistic photographic textures than standard CFG. Each column is a pair generated from the same prompt (shown above): the top row uses standard CFG and the bottom row uses BOG. Across all four prompts, BOG yields stronger photographic realism and finer texture detail while preserving the content.

4 Conclusion, Limitations, and Future Directions

In this report, we present Boogu-Image-0.1, an open-source unified system for image generation and editing that jointly designs understanding and generation as one integrated whole. This design is motivated by a shift we observe in the field: generation is moving from Text-to-Image toward Requirement-to-Image, where the model must interpret complex intentions, implicit constraints, and cross-modal context rather than a single descriptive prompt. The training of Boogu-Image-0.1 is highly cost-effective, enabled by a carefully designed data construction strategy. On top of this, inference-time scaling with an agentic system powered by Deepseek-V4-Flash that rewrites prompts and selects models further improves performance. Comprehensive evaluations demonstrate that Boogu-Image-0.1 has emerged as one of the top open-source models currently available, achieving performance near to leading closed-source models such as NanoBanana-Pro in certain domains. Despite this strong performance, it maintains economical training costs, requiring 208.62M unique images across its full training pipeline at a total cost of around \$400K.

While Boogu achieves competitive performance under limited resources, it still has several limitations. In terms of world knowledge, Boogu still trails leading closed-source systems on tasks demanding extensive common-sense and domain knowledge (e.g., artistic styles, landmarks, public figures, and commercial products), and this gap is hard to measure reliably, so we expect it to exceed what reported scores suggest. For text rendering, only Chinese and English are currently optimized. In terms of body structure, multi-person interactions, heavy occlusions, or atypical viewpoints can yield anatomical inconsistencies such as distorted hands and limbs. Because we adopt the open-source FLUX.1 VAE, its reconstruction error further bottlenecks fine-grained details like small faces, limbs, and text, which native 2K generation only partially alleviates.

With the open-source release of Boogu-Image-0.1, we aim to advance image generation toward systems that genuinely understand what users want. We believe the following directions are essential for the field to advance, and we call on the community to jointly explore them:

- **Toward more transparent and faithful evaluation.** Static benchmarks have largely saturated and grown misleading, as models increasingly optimize for the benchmark itself, so high academic scores no longer guarantee real-world efficacy. Even a model that dominates standard benchmarks can deliver an inferior experience, since true utility lies in faithfully translating complex, multi-faceted intents into images (through multi-turn editability, identity-preserving consistency, and robust adherence to long-context prompts) rather than single-shot aesthetic over-fitting.
- **Overcoming the open-source data bottleneck.** Public datasets suffer from pervasive semantic noise and a lack of cognitive depth, imposing a strict ceiling on performance, and closing the gap to frontier models requires a shift toward high-fidelity data synthesis and expert-level curation.
- **Toward stronger agentic image generation.** Injecting understanding at inference time has proven effective, but current agentic pipelines remain shallow. We see substantial headroom in tighter reasoning-generation loops, self-verification, and tool use, moving generation from a passive mapping toward a system that actively plans, critiques, and refines its own outputs.

References

- [1] Arena ai: The official ai ranking & llm leaderboard. <https://arena.ai/>, 2026.
- [2] Ideogram AI. Ideogram 4. <https://ideogram.ai/blog/ideogram-4.0/>, 2026.
- [3] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibao Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL <https://arxiv.org/abs/2511.21631>.
- [4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL <https://arxiv.org/abs/2502.13923>.
- [5] BIG bench authors. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL <https://openreview.net/forum?id=uyTL5Bvosj>.
- [6] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science*. <https://cdn.openai.com/papers/dall-e-3.pdf>, 2(3):8, 2023.
- [7] Black Forest Labs. Announcing Black Forest Labs: Black forest labs’ inaugural model suite, FLUX.1, 2024. URL <https://blackforestlabs.ai/announcing-black-forest-labs/>. Accessed: 2026-04-25.
- [8] Black Forest Labs. FLUX.2: Analyzing and enhancing the latent space of FLUX – representation comparison, 11 2025. URL <https://bfl.ai/research/representation-comparison>.
- [9] Black Forest Labs. Flux.2: Production-grade ai image generation and editing model with 4mp photorealistic output and multi-reference control. <https://bfl.ai/models/flux-2>, 2025.
- [10] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=B1xsqj09Fm>.
- [11] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.
- [12] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Leo Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.
- [13] Minwoo Byeon, Beomhee Park, Haechon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. <https://github.com/kakaobrain/coyo-dataset>, 2022.
- [14] ByteDance Seed Team. Seedream 4.5. https://seed.bytedance.com/en/seedream4_5, 2026.
- [15] ByteDance Seed Team. Seedream 5.0 lite. <https://seed.bytedance.com/en/blog/deeper-thinking-more-accurate-generation-introducing-seedream-5-0-lite>, 2026.
- [16] Huanqia Cai, Yijun Yang, and Winston Hu. Mm-iq: Benchmarking human-like abstraction and reasoning in multimodal models. *arXiv preprint arXiv:2502.00698*, 2025.
- [17] Qi Cai, Jingwen Chen, Chengmin Gao, Zijian Gong, Yehao Li, Tao Mei, Yingwei Pan, Yi Peng, Zhaofan Qiu, Ting Yao, Kai Yu, Yiheng Zhang, et al. Hidream-o1-image: A natively unified image generative foundation model with pixel-level unified transformer. *arXiv preprint arXiv:2605.11061*, 2026.

- [18] Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, Jun Zhao, et al. Rwku: Benchmarking real-world knowledge unlearning for large language models. Advances in Neural Information Processing Systems, 37:98213–98263, 2024.
- [19] Siyu Cao, Hangting Chen, Peng Chen, Yiji Cheng, Yutao Cui, Xinchu Deng, Ying Dong, Kipper Gong, Tianpeng Gu, Xiusen Gu, et al. Hunyuanimage 3.0 technical report. arXiv preprint arXiv:2509.23951, 2025.
- [20] Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwal, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In 32nd USENIX security symposium (USENIX Security 23), pp. 5253–5270, 2023.
- [21] 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, 2025. URL <https://arxiv.org/abs/2506.07977>.
- [22] Dong Chen, Fangyun Wei, Ziyu Wan, Dongdong Chen, Jiawei Zhang, Jinjing Zhao, Sirui Zhang, Yang Yue, Zhiyang Liang, Baining Guo, et al. Lens: Rethinking training efficiency for foundational text-to-image models. arXiv preprint arXiv:2605.21573, 2026.
- [23] Jiuhaì Chen, Le Xue, Zhiyang Xu, Xichen Pan, Shusheng Yang, Can Qin, An Yan, Honglu Zhou, Zeyuan Chen, Lifu Huang, Tianyi Zhou, Junnan Li, Silvio Savarese, Caiming Xiong, and Ran Xu. Blip3o-next: Next frontier of native image generation, 2025. URL <https://arxiv.org/abs/2510.15857>.
- [24] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart- α : Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023.
- [25] Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance, 2023. URL <https://arxiv.org/abs/2305.05176>.
- [26] Ziliang Chen, Xin Huang, Xiaoxuan Fan, Keze Wang, Yuyu Zhou, Quanlong Guan, and Liang Lin. Reproducible vision-language models meet concepts out of pre-training. In Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 14701–14711, 2025.
- [27] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. arXiv preprint arXiv:2403.04132, 2024.
- [28] K Alec Chrystal and Paul D Mizen. Goodhart’s law: its origins, meaning and implications for monetary policy. Central banking, monetary theory and practice: Essays in honour of Charles Goodhart, 1:221–243, 2003.
- [29] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin F. Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Patrick Collier, Alexey Gritsenko, Vignesh Birodkar, Cristina Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetić, Dustin Tran, Thomas Kipf, Mario Lučić, Xiaohua Zhai, Daniel Keysers, Jeremiah Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters, 2023. URL <https://arxiv.org/abs/2302.05442>.
- [30] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255. Ieee, 2009.
- [31] Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks Lakshmanan, and Ahmed H Awadallah. Hybrid llm: Cost-efficient and quality-aware query routing. In International Conference on Learning Representations, volume 2024, pp. 41348–41366, 2024.
- [32] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), Advances in Neural Information Processing Systems, volume 34, pp. 19822–19835. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/a4d92e2cd541fca87e4620aba658316d-Paper.pdf.

- [33] Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 1286–1305, 2021.
- [34] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. URL <https://openreview.net/forum?id=YicbFdNTTy>.
- [35] 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 e-prints*, pp. arXiv–2503, 2025.
- [36] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning, ICML’24*. JMLR.org, 2024.
- [37] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruva Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: in search of the next generation of multimodal datasets. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS ’23*, Red Hook, NY, USA, 2023. Curran Associates Inc.
- [38] Peng Gao, Le Zhuo, Dongyang Liu, Ruoyi Du, Xu Luo, Longtian Qiu, Yuhang Zhang, Rongjie Huang, Shijie Geng, Renrui Zhang, Junlin Xie, Wenqi Shao, Zhengkai Jiang, Tianshuo Yang, Weicai Ye, Tong He, Jingwen He, Junjun He, Yu Qiao, and Hongsheng Li. Lumina-t2x: Scalable flow-based large diffusion transformer for flexible resolution generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=EbWf36quzd>.
- [39] Yu Gao, Lixue Gong, Qiushan Guo, Xiaoxia Hou, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, Wei Liu, Yichun Shi, Shiqi Sun, Yu Tian, Zhi Tian, Peng Wang, Rui Wang, Xuanda Wang, Xun Wang, Ye Wang, Guofeng Wu, Jie Wu, Xin Xia, Xuefeng Xiao, Zhonghua Zhai, Xinyu Zhang, Qi Zhang, Yuwei Zhang, Shijia Zhao, Jianchao Yang, and Weilin Huang. Seedream 3.0 technical report, 2025. URL <https://arxiv.org/abs/2504.11346>.
- [40] Google Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL <https://arxiv.org/abs/2507.06261>.
- [41] Zigang Geng, Yibing Wang, Yeyao Ma, Chen Li, Yongming Rao, Shuyang Gu, Zhao Zhong, Qinglin Lu, Han Hu, Xiaosong Zhang, et al. X-omni: Reinforcement learning makes discrete autoregressive image generative models great again. *arXiv preprint arXiv:2507.22058*, 2025.
- [42] Dhruva 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.
- [43] Google. Introducing nano banana pro, 2025. URL <https://blog.google/innovation-and-ai/products/nano-banana-pro/>.
- [44] Google. Nano banana 2: Combining pro capabilities with lightning-fast speed, 2026. URL <https://blog.google/innovation-and-ai/technology/ai/nano-banana-2/>.
- [45] Google DeepMind. Imagen 4.0. <https://deepmind.google/models/imagen/>, 2025.
- [46] Google DeepMind. Gemini 3.1 pro - model card. <https://deepmind.google/models/model-cards/gemini-3-1-pro/>, February 2026. Accessed: 2026-05-05.
- [47] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour, 2018. URL <https://arxiv.org/abs/1706.02677>.

- [48] Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. <https://github.com/huggingface/accelerate>, 2022.
- [49] Meng-Hao Guo, Jiajun Xu, Yi Zhang, Jiayi Song, Haoyang Peng, Yi-Xuan Deng, Xinzhi Dong, Kiyohiro Nakayama, Zhengyang Geng, Chen Wang, et al. R-bench: Graduate-level multi-disciplinary benchmarks for llm & mllm complex reasoning evaluation. [arXiv preprint arXiv:2505.02018](#), 2025.
- [50] Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization, 2018. URL <https://arxiv.org/abs/1802.09568>.
- [51] Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. Optimizing prompts for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:66923–66939, 2023.
- [52] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [53] Zefeng He, Siyuan Huang, Xiaoye Qu, Yafu Li, Tong Zhu, Yu Cheng, and Yang Yang. Gems: Agent-native multimodal generation with memory and skills. [arXiv preprint arXiv:2603.28088](#), 2026.
- [54] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.
- [55] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL <https://openreview.net/forum?id=qw8AKxfYbI>.
- [56] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS ’20*, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.
- [57] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. [arXiv preprint arXiv:2403.05135](#), 2024.
- [58] Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A Smith. Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20406–20417, 2023.
- [59] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36: 78723–78747, 2023.
- [60] Zijian Huang, Jay Zhangjie Wu, Zian Wang, Tianshi Cao, Jiasi Chen, Sanja Fidler, Huan Ling, and Xuanchi Ren. Ape: Agentic prompt enhancer for image generation and editing. [arXiv preprint arXiv:2606.00204](#), 2026.
- [61] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.
- [62] Zhangqi Jiang, Zheng Sun, Xianfang Zeng, Yufeng Yang, Xuanyang Zhang, Yongliang Wu, Wei Cheng, Gang Yu, Xu Yang, and Bihan Wen. Geditbench v2: A human-aligned benchmark for general image editing. [arXiv preprint arXiv:2603.28547](#), 2026.
- [63] Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL <https://kellerjordan.github.io/posts/muon/>.
- [64] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. [arXiv preprint arXiv:2001.08361](#), 2020.
- [65] Tero Karras, Miika Aittala, Samuli Laine, and Timo Aila. Elucidating the design space of diffusion-based generative models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS ’22*, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

- [66] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models, 2024. URL <https://arxiv.org/abs/2312.02696>.
- [67] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [68] 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.
- [69] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [70] Kuaishou Technology. Kling-image-2.1. https://kling.ai/explore/kling_2.1_api, 2025.
- [71] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.
- [72] Sangwu Lee, Erwann Millon, Le Zhuo, Matthew Newton, Andrei Filatov, Naga Sai Abhinay Devarinti, Dazhi Zhong, Avram Djordjevic, Gabriel Menezes, Will Beddow, Titus Ebbecke, Mihai Petrescu, Owen Fahey, Gian Saß, Felix Gil, and Victor Perez. Krea 2. <https://www.krea.ai/blog/krea-2-technical-report>, 2026.
- [73] Niantong Li, Guangzheng Hu, Weixu Qiao, Ying Ba, Qichen Hong, Shijun Shen, Jinlin Wang, Fan Zhou, Jianye Kang, Xin Shang, Ziyi He, Wei Wang, Dalin Li, Jiahao Li, Jie Zhang, Kaiyuan Gao, Kun Yan, Lihan Jiang, Ningyuan Tang, Shengming Yin, Tianhe Wu, Xiao Xu, Xiaoyue Chen, Yuxiang Chen, Yan Shu, Yanran Zhang, Yilei Chen, Yixian Xu, Zekai Zhang, Zhendong Wang, Zihao Liu, Zikai Zhou, Hongzhu Shi, Yi Wang, Bing Zhao, Hu Wei, Lin Qu, and Chenfei Wu. Qwen-image-bench: From generation to creation in text-to-image evaluation, 2026. URL <https://arxiv.org/abs/2605.28091>.
- [74] Qingyun Li, Zhe Chen, Weiyun Wang, Wenhai Wang, Shenglong Ye, Zhenjiang Jin, Guanzhou Chen, Yinan He, Zhangwei Gao, Erfei Cui, Jiashuo Yu, Hao Tian, Jiasheng Zhou, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Pinlong Cai, Licheng Wen, Xiangchao Yan, Pei Chu, Yi Wang, Min Dou, Changyao Tian, Xizhou Zhu, Lewei Lu, Yushi Chen, Junjun He, Tong Lu, Yali Wang, Limin Wang, Dahua Lin, Yu Qiao, Botian Shi, Conghui He, and Jifeng Dai. Omnicorpus: A unified multimodal corpus of 10 billion-level images interleaved with text. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=kwqhn2VuG4>.
- [75] Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Arsh Koneru, Yusuke Kato, Kazuki Kozuka, and Aditya Grover. Reflect-dit: Inference-time scaling for text-to-image diffusion transformers via in-context reflection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15657–15668, 2025.
- [76] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.
- [77] Zhengyang Liang, Hao He, Ceyuan Yang, and Bo Dai. Scaling laws for diffusion transformers. *arXiv preprint arXiv:2410.08184*, 2024.
- [78] Jintao Lin, Bowen Dong, Weikang Shi, Chenyang Lei, Suiyun Zhang, Rui Liu, and Xihui Liu. Aegis: Exploring the limit of world knowledge capabilities for unified mulitmodal models. *arXiv preprint arXiv:2601.00561*, 2026.
- [79] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL <https://arxiv.org/abs/2210.02747>.
- [80] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.
- [81] Jiaxiang Liu, Zhida Feng, Pengyu Zou, Zhenyu Qian, Tianrui Zhu, Jun Xia, Yuehu Dong, Yanzheng Lin, Honglin Xiong, Anqi Chen, et al. Ernie-image technical report. *arXiv preprint arXiv:2605.25347*, 2026.
- [82] 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, 2025. URL <https://arxiv.org/abs/2505.05470>.
- [83] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

- [84] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. URL <https://arxiv.org/abs/2209.03003>.
- [85] 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, 2022. URL <https://arxiv.org/abs/2206.00927>.
- [86] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22(4):730–751, June 2025. ISSN 2731-5398. doi: 10.1007/s11633-025-1562-4. URL <http://dx.doi.org/10.1007/s11633-025-1562-4>.
- [87] Shiyin Lu, Yang Li, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Han-Jia Ye. Ovis: Structural embedding alignment for multimodal large language model. *arXiv preprint arXiv:2405.20797*, 2024.
- [88] Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. URL <http://probml.github.io/book2>.
- [89] Johann von Neumann. Some matrix-inequalities and metrization of matric-space. *Tomsk. Univ. Rev.*, 1:286–300, 1937. Reprinted in *John von Neumann: Collected Works* (A. H. Taub, ed.), Vol. IV, Pergamon Press, Oxford, 1962, pp. 205-218.
- [90] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.
- [91] Yuwei Niu, Munan Ning, Mengren Zheng, Weiyang Jin, Bin Lin, Peng Jin, Jiaqi Liao, Kunpeng Ning, Chaoran Feng, Bin Zhu, and Li Yuan. Wise: A world knowledge-informed semantic evaluation for text-to-image generation. *arXiv preprint arXiv:2503.07265*, 2025.
- [92] Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data. *arXiv preprint arXiv:2406.18665*, 2024.
- [93] OpenAI. Gpt image 1. <https://developers.openai.com/api/docs/models/gpt-image-1>, 2025.
- [94] OpenAI. Introducing chatgpt images 2.0, 2026. URL <https://openai.com/index/introducing-chatgpt-images-2-0/>.
- [95] Shubham Parashar, Zhiqiu Lin, Tian Liu, Xiangjue Dong, Yanan Li, Deva Ramanan, James Caverlee, and Shu Kong. The neglected tails in vision-language models, 2024. URL <https://arxiv.org/abs/2401.12425>.
- [96] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11410–11420, 2022.
- [97] William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL <https://arxiv.org/abs/2212.09748>.
- [98] 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 *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024.
- [99] Qwen Team, Alibaba Group. Qwen-image-2512: Finer details, greater realism. <https://qwen.ai/blog?id=qwen-image-2512>, 2025. Apache-2.0 License.
- [100] Qwen Team, Alibaba Group. Qwen-image-edit-2509: Multi-image support, improved consistency. <https://qwen.ai/blog?id=7a90090115ee193ce6a7f619522771dd9696dd93>, 2025. Apache-2.0 License.
- [101] Qwen Team, Alibaba Group. Qwen-image-2511. <https://qwen.ai/blog?id=qwen-image-2512>, 2025. Apache-2.0 License.
- [102] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.
- [103] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models, 2020. URL <https://arxiv.org/abs/1910.02054>.
- [104] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

- [105] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pp. 5389–5400. PMLR, 2019.
- [106] Ehud Reiter. A structured review of the validity of bleu. *Computational Linguistics*, 44(3):393–401, 2018.
- [107] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL <https://arxiv.org/abs/2112.10752>.
- [108] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241. Springer, 2015.
- [109] Seyedmorteza Sadat, Otmar Hilliges, and Romann M. Weber. Eliminating oversaturation and artifacts of high guidance scales in diffusion models, 2025. URL <https://arxiv.org/abs/2410.02416>.
- [110] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.
- [111] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.
- [112] 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.
- [113] Team Seedream, :, Yunpeng Chen, Yu Gao, Lixue Gong, Meng Guo, Qiushan Guo, Zhiyao Guo, Xiaoxia Hou, Weilin Huang, Yixuan Huang, Xiaowen Jian, Huafeng Kuang, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, Wei Liu, Yanzuo Lu, Zhengxiong Luo, Tongtong Ou, Guang Shi, Yichun Shi, Shiqi Sun, Yu Tian, Zhi Tian, Peng Wang, Rui Wang, Xun Wang, Ye Wang, Guofeng Wu, Jie Wu, Wenxu Wu, Yonghui Wu, Xin Xia, Xuefeng Xiao, Shuang Xu, Xin Yan, Ceyuan Yang, Jianchao Yang, Zhonghua Zhai, Chenlin Zhang, Heng Zhang, Qi Zhang, Xinyu Zhang, Yuwei Zhang, Shijia Zhao, Wenliang Zhao, and Wenjia Zhu. Seedream 4.0: Toward next-generation multimodal image generation, 2025. URL <https://arxiv.org/abs/2509.20427>.
- [114] Vasu Singla, Kaiyu Yue, Sukriti Paul, Reza Shirkavand, Mayuka Jayawardhana, Alireza Ganjdaneh, Heng Huang, Abhinav Bhatele, Gowthami Somepalli, and Tom Goldstein. From pixels to prose: A large dataset of dense image captions, 2024. URL <https://arxiv.org/abs/2406.10328>.
- [115] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. URL <https://arxiv.org/abs/1503.03585>.
- [116] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6048–6058, 2023.
- [117] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL <https://arxiv.org/abs/2010.02502>.
- [118] Lin Song, Wenbo Li, Guoqing Ma, Wei Tang, Bo Wang, Yuan Zhang, Yijun Yang, Yicheng Xiao, Jianhui Liu, Yanbing Zhang, et al. Joyai-image: Awaking spatial intelligence in unified multimodal understanding and generation. *arXiv preprint arXiv:2605.04128*, 2026.
- [119] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. URL <https://arxiv.org/abs/2011.13456>.
- [120] Stability AI. Introducing stable diffusion 3.5. <https://stability.ai/news/introducing-stable-diffusion-3-5>, Oct 2024.
- [121] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024.

- [122] 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.
- [123] Super Intelligence Team, Changhao Qiao, Chao Hui, Chen Li, Cunzheng Wang, Dejia Song, Jiale Zhang, Jing Li, Qiang Xiang, Runqi Wang, et al. Firered-image-edit-1.0 technical report. [arXiv preprint arXiv:2602.13344](#), 2026.
- [124] Vishaal Udandarao, Ameya Prabhu, Adhiraj Ghosh, Yash Sharma, Philip H. S. Torr, Adel Bibi, Samuel Albanie, and Matthias Bethge. No "zero-shot" without exponential data: Pretraining concept frequency determines multimodal model performance, 2024. URL <https://arxiv.org/abs/2404.04125>.
- [125] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
- [126] Aowen Wang, Wei Li, Hao Luo, Mengxing Ao, Chenyu Zhu, Xinyang Li, and Fan Wang. Jco-mvton: Jointly controllable multi-modal diffusion transformer for mask-free virtual try-on. [arXiv preprint arXiv:2508.17614](#), 2025.
- [127] Guo-Hua Wang, Shanshan Zhao, Xinjie Zhang, Liangfu Cao, Pengxin Zhan, Lunhao Duan, Shiyin Lu, Minghao Fu, Xiaohao Chen, Jianshan Zhao, Yang Li, and Qing-Guo Chen. Ovis-u1 technical report, 2025. URL <https://arxiv.org/abs/2506.23044>.
- [128] Ross Wightman. Pytorch image models. <https://github.com/rwightman/pytorch-image-models>, 2019.
- [129] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm, 2021. URL <https://arxiv.org/abs/2110.00476>.
- [130] 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, Shutong Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. URL <https://arxiv.org/abs/2508.02324>.
- [131] Chenyuan Wu, Pengfei Zheng, Ruirao Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ze Liu, Ziyi Xia, Chaofan Li, Haoge Deng, Jiahao Wang, Kun Luo, Bo Zhang, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. Omnigen2: Exploration to advanced multimodal generation, 2025. URL <https://arxiv.org/abs/2506.18871>.
- [132] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruirao Yan, Chaofan Li, Shutong Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13294–13304, 2025.
- [133] 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 image synthesis with linear diffusion transformer, 2024. URL <https://arxiv.org/abs/2410.10629>.
- [134] Jiazhen Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.
- [135] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL <https://arxiv.org/abs/2505.09388>.
- [136] Yang Ye, Xianyi He, Zongjian Li, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, Li Yuan, et al. Imgedit: A unified image editing dataset and benchmark. *Advances in Neural Information Processing Systems*, 38, 2026.

- [137] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.
- [138] Z-Image Team, 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, 2025. URL <https://arxiv.org/abs/2511.22699>.
- [139] Z.AI. Glm-image: Auto-regressive for dense-knowledge and high-fidelity image generation. <https://z.ai/blog/glm-image>, 2026. Apache-2.0 License.
- [140] Daoan Zhang, Che Jiang, Ruoshi Xu, Biaoxiang Chen, Zijian Jin, Yutian Lu, Jianguo Zhang, Liang Yong, Jiebo Luo, and Shengda Luo. Worldgenbench: A world-knowledge-integrated benchmark for reasoning-driven text-to-image generation. *arXiv preprint arXiv:2505.01490*, 2025.
- [141] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023.
- [142] Zekai Zhang, Jiahao Li, Jie Zhang, Kaiyuan Gao, Kun Yan, Lihan Jiang, Ningyuan Tang, Shengming Yin, Tianhe Wu, Xiaoyue Chen, Xiao Xu, Yan Shu, Yanran Zhang, Yixian Xu, Yuxiang Chen, Zhendong Wang, Zihao Liu, Zikai Zhou, Huishuai Zhang, Dongyan Zhao, and Chenfei Wu. Qwen-image-agent: Bridging the context gap in real-world image generation, 2026. URL <https://arxiv.org/abs/2606.26907>.
- [143] Bing Zhao, Chenfei Wu, Deqing Li, Hao Meng, Jiahao Li, Jie Zhang, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kuan Cao, et al. Qwen-image-2.0 technical report. *arXiv preprint arXiv:2605.10730*, 2026.
- [144] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. Pytorch fsdp: Experiences on scaling fully sharded data parallel, 2023. URL <https://arxiv.org/abs/2304.11277>.
- [145] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhi Zheng Lin, Zi Ning Li, Dacheng Li, Eric Xing, et al. Chatbot arena: An open platform for evaluating llms based on human preference. In *arXiv preprint arXiv:2403.04132*, 2024.
- [146] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingting Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhui Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025. URL <https://arxiv.org/abs/2504.10479>.
- [147] Le Zhuo, Ruoyi Du, Han Xiao, Yangguang Li, Dongyang Liu, Rongjie Huang, Wenze Liu, Xiangyang Zhu, Fu-Yun Wang, Zhanyu Ma, Xu Luo, Zehan Wang, Kaipeng Zhang, Lirui Zhao, Si Liu, Xiangyu Yue, Wanli Ouyang, Yu Qiao, Hongsheng Li, and Peng Gao. Lumina-next: making lumina-t2x stronger and faster with next-dit. In *Proceedings of the 38th International Conference on Neural Information Processing Systems, NIPS '24*, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.
- [148] 国家语言文字工作委员会. 汉字处. *现代汉语常用字表*. 语文出版社, 1988. URL https://books.google.com.hk/books?id=Av_tAQAAACAAJ.

Appendix

A Contributors

Boogu Team

Celia Large Model Application Laboratory (Leibniz)



- **Project Sponsors:** Rui Liu¹, Chao Huang², Han Shi¹, Haoli Bai¹, Xihui Liu², Hongsheng Li⁶, Qifeng Chen³
- **Project Leader:** Chenyang Lei¹
- **Core Contributors:** Guoxuan Chen^{1,2,*}, Chufeng Xiao¹, Haoran Yang¹, Siyue Xie¹, Binxiao Huang¹, Ming Zhang¹, Cheuk Him Chau¹, Xinyu Fu¹, Yingzhao Lian¹, Tom S.Y. Li¹, Jintao Lin^{1,2,*}, Bowen Dong^{1,4,*}, Zian Qian^{1,3,*}, Chenyang Lei¹
- **Contributors:** Yuhao Liu^{1,5,*}, Yuxuan Hu^{1,6,*}, Weikang Shi^{1,6,*}, Bin Zou^{1,2,*}, Bowen Zheng^{1,7,*}, Haoxuan Che^{1,†}, Chang Chen^{1,3,*}, Yuyang He¹, Heyang Sun^{1,3,*}, Tianyu Huang^{1,4,*}, Chong Hou Choi¹, Cheng Gong¹
- **Correspondence:** Chenyang Lei¹, Rui Liu¹, Chao Huang²

Primary Affiliation: ¹Artificial Intelligence Laboratory (Leibniz), Huawei Technologies, Hong Kong.



Collaborating Institutions: ²The University of Hong Kong, ³The Hong Kong University of Science and Technology, ⁴The Hong Kong Polytechnic University, ⁵City University of Hong Kong, ⁶The Chinese University of Hong Kong, ⁷Nanjing University.

*Note: * indicates work completed during an internship. † indicates the author is no longer affiliated with the primary affiliation.*

B Background and Technical Choices

B.1 Background

The evolution of visual synthesis has rapidly transitioned from isolated generative tasks to integrated multimodal frameworks. This section reviews the foundational trajectories contextualizing our work: text-to-image diffusion, controllable editing, vision-language models (VLMs), and the recent paradigm shift towards unified generation.

Text-to-Image Diffusion Models. Diffusion models [56] have fundamentally revolutionized visual synthesis. Foundational works like LDM [107] and DALL-E 2 [104] established the paradigm of high-fidelity text-conditioned generation. Subsequent advancements focused on text-image alignment, with DALL-E 3 [6] leveraging recaptioned datasets for superior prompt adherence. Furthermore, the architectural shift from U-Nets [108] to transformers [125], pioneered by DiT [97], unlocked unprecedented scalability. This transformer-driven scalability is prominently demonstrated by the spatiotemporal video generation of Sora [12] and the rectified flow formulations of Stable Diffusion 3 [36] for complex image synthesis. Most recently, this trajectory has culminated in the proprietary and closed-source Nano Banana series [43] and ChatGPT Images series [94], which currently define the state-of-the-art in high-fidelity text-to-image generation.

Controllable Generation and Editing. Relying solely on text prompts often falls short when precise spatial control is required. To address this, ControlNet [141] introduced auxiliary conditioning, enabling diffusion models to respect spatial hints (e.g., depth or poses) without compromising the original generative prior. In the domain of image editing, InstructPix2Pix [11] pioneered instruction-driven modifications, allowing users to alter existing images via natural language instructions without relying on complex manual masks.

Vision-Language Models (VLMs). Driven by the success of LLMs, VLMs have rapidly advanced visual comprehension. Pioneering works like LLaVA [80] aligned visual encoders—predominantly based on Vision Transformers (ViTs) [34]—with LLMs via visual instruction tuning. Recent state-of-the-art models, such as Ovis [87] and the Qwen-VL series [3, 4], have further refined structural alignment and visual reasoning, enabling the processing of high-resolution inputs for intricate multimodal tasks.

Towards Unified Multimodal Generation. Building upon VLMs’ reasoning and diffusion models’ generative prowess, the community is shifting towards consolidated architectures. OmniGen [132] and OmniGen2 [131] pioneered this by natively handling diverse generation and editing tasks within a single framework. Concurrently, while proprietary models like Google’s Nano Banana series [43] have introduced reasoning-guided synthesis via interleaved prompts, open-weight alternatives such as the Qwen-Image series [130] and Z-Image [138] integrate deep visual understanding with scalable, efficient generation, paving the way for versatile and intuitive multimodal AI systems.

B.2 Macro-Architecture Design

Empirical observations of state-of-the-art proprietary models, such as ChatGPT Images series [94] and the Nano Banana series [43], reveal a consistent latency prior to generation. We hypothesize that, regardless of their underlying architectures, these systems employ a robust comprehension module to meticulously analyze and deconstruct user instructions, which naturally incurs a temporal overhead. Motivated by this, our preliminary architectural design utilized a massive 72B Vision-Language Model (VLM) to drive a 10B Diffusion Transformer (DiT), where the VLM was kept frozen and only the DiT parameters were updated. However, this configuration imposes prohibitive computational overheads during both training and inference, rendering it inaccessible to the broader research community. To address this, we propose a decoupled, compute-efficient paradigm: we employ a moderately large VLM (e.g., 32B) as an *Instruction Reasoner* for complex logic comprehension, paired with a lightweight VLM (e.g., 8B) acting as an *Instruction Encoder* to extract hidden states as conditioning signals for the DiT. This decoupled strategy significantly mitigates both training and inference costs compared to relying on a monolithic, large VLM.

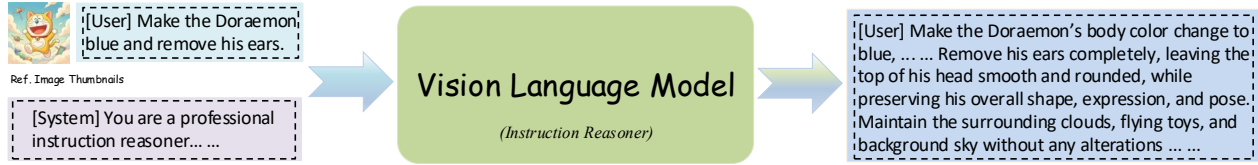


Figure 34 Illustration of the Instruction Reasoner workflow. The input text instruction, formatted via a predefined chat template with a system prompt, is processed by the Reasoner. In image editing scenarios, a reference image is appended to the input. The Reasoner yields a comprehensively analyzed, deconstructed, and rewritten instruction to condition the subsequent generation pipeline.

B.2.1 Instruction Reasoner

The operational pipeline of the Instruction Reasoner is depicted in Figure 34. We introduce a unified architecture that seamlessly integrates both text-to-image (T2I) generation and image editing. Specifically, the system determines the task modality based on the presence of visual inputs: if the user-provided reference image is omitted, the process defaults to a T2I task and applies the corresponding T2I system prompt. Conversely, the inclusion of a reference image explicitly designates an editing operation, thereby activating an editing-specific system prompt. Ultimately, the Reasoner outputs a meticulously parsed, structurally decomposed, and refined textual prompt, which serves as the optimized conditioning signal for the subsequent generation stages.

B.2.2 Instruction Encoder and Image Generation

Upon acquiring the text instructions that have been analyzed, deconstructed, and rewritten by the Instruction Reasoner, we employ a subsequent pipeline to generate images. This pipeline primarily consists of two core modules: an Autoregressive Transformer [125] and a Diffusion Transformer (DiT) [97]. The Autoregressive Transformer essentially functions as a Vision-Language Model (VLM), serving as our Instruction Encoder. The sub-modules comprising the pipeline of the Instruction Encoder are briefly introduced as follows (see Figure.35):

Text Tokenizer and Embedding Table This module is responsible for tokenizing the rewritten text instructions and projecting them into a sequence of embedding vectors. This sequence serves as a partial input to the Autoregressive Transformer.

Vision Transformer (ViT) The number of user-provided reference images can range from zero to multiple. When present, the Vision Transformer (ViT) [34] encodes them into the same latent space as the text prompt embeddings. These visual and textual embeddings are concatenated into a unified sequence before being fed into the Autoregressive Transformer. As a standard component of most VLMs, the ViT processes these reference images, which are here downsampled to a 384×384 resolution to prevent excessively long vector sequences. Notably, if no reference images are provided, the task is defined as pure text-to-image generation, and the ViT is consequently bypassed.

System Prompt The task paradigm is determined based on the presence of reference images. Specifically, the absence of any reference images defines a text-to-image generation task, whereas the presence of one or more reference images indicates an image editing task. The system prompt is adaptively formulated corresponding to the specific task type.

Prompt Tuning Transformer (Optional) This is a highly lightweight, trainable Transformer module. In our implementation, it is fixed at 3 layers, with the architecture of each layer illustrated in Figure 37. It contains only 32 trainable prompt token embeddings and employs causal masking. The output from its final layer is prepended to the input sequence of the Instruction Encoder. The Prompt Tuning Transformer serves as an optional trainable module designed to enhance the instruction parsing and comprehension capabilities of the overall text-to-image model. Since the Instruction Encoder remains frozen during training, mathematically, the introduction of this module globally modulates the embedding distribution of the hidden state sequence output by the encoder, thereby facilitating a better alignment with the semantic space required by the DiT.

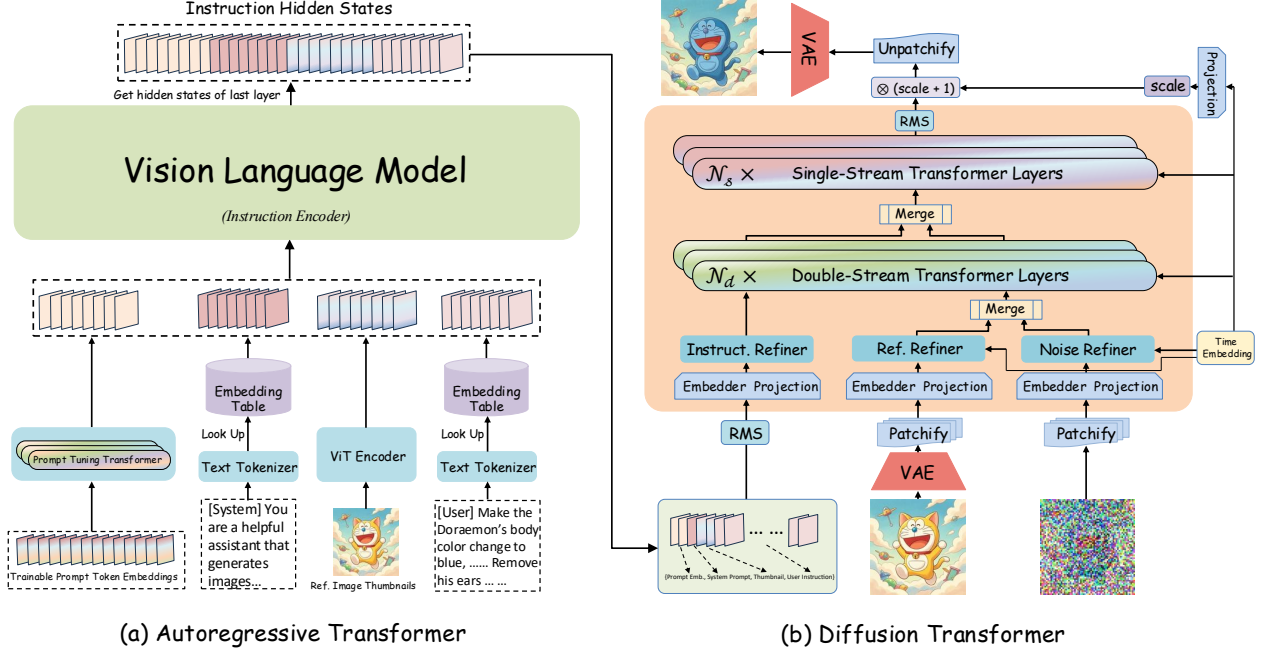


Figure 35 Pipeline of the instruction encoder and DiT. The framework defaults to image editing when a reference image is provided, and text-to-image generation otherwise. A thumbnail of the reference image is fed to the Instruction Encoder to aid comprehension, while the original image (scaled within the resolution limit) is processed by the DiT. The system prompt adaptively adjusts to the specific task. Furthermore, an optional trainable Prompt Tuning Transformer is introduced to enhance instruction parsing. Note that the input user instructions have been pre-analyzed and rewritten by the Instruction Reasoner.

It is worth noting that the Prompt Tuning Transformer is an optional module, which will be discussed further in the subsequent experimental section. In addition, Figure 35(b) illustrates the Diffusion Transformer (DiT) component. The sub-modules involved in this component are briefly introduced below:

Variational Autoencoder (VAE) When user-provided reference images are present, to balance performance and computational efficiency, we first employ a Variational Autoencoder (VAE) [67] encoder to map them into a latent space before feeding them into the DiT. Consequently, our diffusion training is conducted entirely within this latent space [107]. Once mapped, the reference images undergo patchification, followed by processing through an Embedder and a Refiner. The resulting hidden states, along with the hidden states corresponding to the noise, are fed into the DiT layers for sequential processing. The output of the final layer must be unpatchified and passed through the VAE decoder to reconstruct the output image. When no reference images are provided, the VAE encoder is bypassed; however, the VAE decoder remains essential for decoding the final output image, as the noise depicted in the bottom right of Figure 35(b) essentially represents the noise state within the VAE-encoded latent space.

Embedder and Refiner The Embedder is a standard linear layer designed to project hidden states from one latent space to another. The Refiner is a simple, lightweight Transformer [125]. By default, it consists of only 2 layers, with each layer’s architecture depicted in Figure 37 (identical to the structure of the Prompt Tuning Transformer). Compared to the Embedder, the Refiner also accounts for the inter-relationships among elements in the newly patchified sequence. Notably, the user instructions, reference images, and noise each possess their own independent Embedders and Refiners.

Single-Stream and Dual-Stream Layers Our DiT employs a stacked architecture comprising $\mathcal{N}_d$ dual-stream layers followed by $\mathcal{N}_s$ single-stream layers. After being projected into a unified latent space by their respective Refiners and associated modules, the reference images and noise are concatenated and fed into the image stream of the dual-stream layers. Simultaneously, the hidden states of the user instructions from the final layer of the Instruction Encoder are projected into the instruction latent space by their corresponding

Refiner, etc., and then input into the instruction stream of the dual-stream layers. The outputs from the two streams of the final dual-stream layer are concatenated and fed into the unified stream of the single-stream layers. Following a series of operations, the output from the single-stream layers is ultimately decoded by the VAE decoder to produce the final image.

Figure 35 illustrates the macro-architecture design and overall pipeline for text-to-image generation and image editing. During the actual inference phase, the DiT component typically requires multiple iterative steps to synthesize high-fidelity images [56, 97]. This iterative process is accompanied by the transition of the time step t from 0 to 1 (or from 1 to 0, depending on the specific formulation), along with corresponding updates to the Time Embedding. For the specific iterative sampling algorithm during generation, one can employ the standard first-order Euler method commonly utilized in Flow Matching [79], or fast dedicated ODE solvers such as DPM-Solver++ [86].

B.3 Micro-Architecture Details

Building upon the macro-level overview presented in the previous section, we now delve into the micro-architectural details of our model. Our specific design draws inspiration from recent foundational works [24, 29, 36, 38, 79, 97, 120, 127, 130–132, 147], which we have appropriately modified and extended to suit our objectives. Overall, we propose a unified framework that seamlessly supports both text-to-image generation and image editing. By integrating and adapting the aforementioned designs, we incorporate specialized modules to enhance fine-grained image comprehension and comprehensive multimodal alignment. The detailed components are elaborated below.

B.3.1 Dual-Stream Layer

The architecture of the dual-stream layer is illustrated in Figure 36(a). In this design, the user instruction stream and the image stream (comprising the noisy latent and the optional reference image) are processed separately within the layer. Unlike conventional dual-stream architectures, which typically either maintain independent linear projections solely for the Query, Key, and Value tensors of the two streams [126], or decouple only a small subset of additional computational modules [36], our approach draws inspiration from recent advancements [38, 127, 147]. Specifically, it incorporates a more comprehensive set of independent intra-stream and inter-stream modules to facilitate deeper computational processing and feature interaction.

Boosting Multimodal Understanding Profound comprehension of user instructions is crucial for image generation and editing. Our VLM-based Instruction Reasoner deconstructs user prompts, feeding rewritten text instructions to the Instruction Encoder to extract generation conditions. During this process, although this encoder utilizes a ViT to process optional reference images, its primary focus remains text instruction modeling rather than comprehensive multimodal understanding.

To explicitly enhance multimodal understanding, following recent architectures [38, 97, 127, 130, 147], we adopt an independent attention module for the image modality (comprising the noisy latent and optional reference image), as shown in Figure 36(a, right). Within the dual-stream layer, the right stream processes image token interactions to deepen visual comprehension. Conversely, the left stream fuses the instruction and image modalities (Figure 36(a, left)). By incorporating instruction residual connections, it inherently emphasizes instruction modeling from a holistic multimodal perspective, despite performing full-modality attention. Subsequently, the image information derived from the left stream’s full-modality analysis is fused with the residual and internal self-attention outputs of the right stream. This integration yields a refined representation grounded in holistic comprehension while prioritizing the image modality. Furthermore, the dual-stream layer features a combination of shared and independent computational modules, including three independent Query, Key, and Value (QKV) projections. These QKV sets are allocated to: the text modality, the image modality used for holistic enhancement, and the image modality’s internal self-attention, respectively.

Collectively, these two streams—rooted in holistic understanding but respectively emphasizing the instruction and image modalities—enhance the model’s multimodal comprehension at a fine-grained level. This synergy paves the way for more macroscopic and comprehensive modeling in the subsequent single-stream layers after the streams converge.

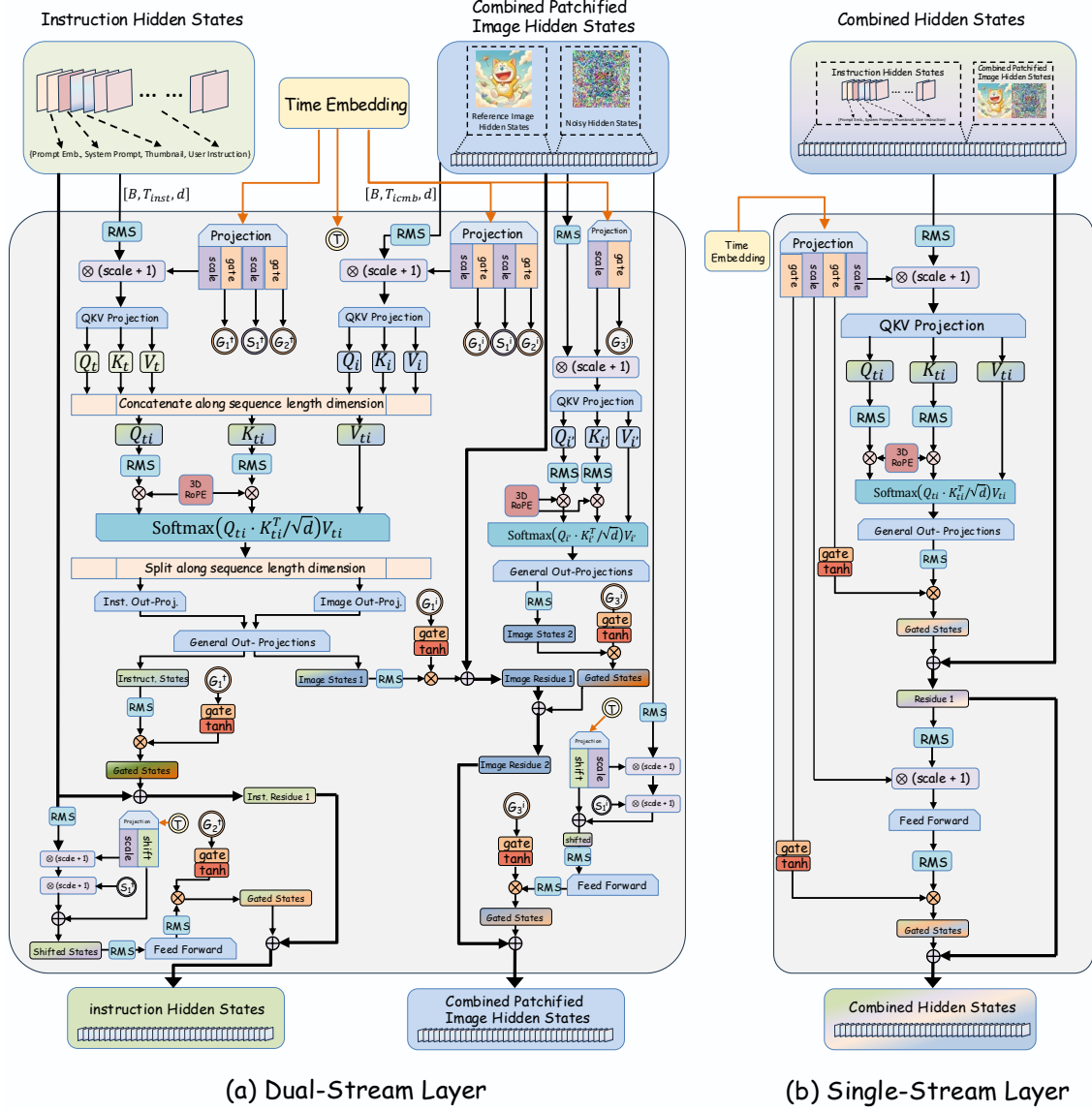


Figure 36 Detailed architecture of the dual-stream and single-stream layers. Bold lines indicate the Main Path constructed via residual connections, while thin lines represent Branch Paths involving complex computations. Retaining only the bold lines reduces both layers to identity mappings. The branch paths merge into the main path at appropriate junctures to influence the parameter distribution of a layer’s output.

Main Path, Residue, and Modulation Our dual-stream layer adopts the classic sandwich norm [32, 147] and the modulation mechanism [97] to facilitate information fusion and signal control. However, our specific implementation introduces slight modifications (see Figure 36(a)). The multi-norm design significantly enhances model stability during pre-training with large learning rates, thereby accelerating convergence and improving overall robustness. In Figure 36(a), the bold lines denote the *Main Path*, which is fundamentally constructed with residual connections to stabilize the parameter distribution across layers. If only this main path existed, a layer reduces to an identity mapping. Conversely, the thin lines represent *Branch Paths* that incorporate various operations, including attention and modulation. These branches are responsible for analyzing and processing the input representations. The computational results from these branch paths are subsequently merged into the main path at appropriate junctures, ultimately shaping the final output distribution of the layer.

B.3.2 Single-Stream Layer

The architecture of our single-stream layer follows numerous classic designs [38, 127, 130, 131, 147]. Specifically, we adopt the classical modulation mechanism [97] and sandwich norm [32, 147] (see Figure 36(a)). This layer merges the hidden states of both the instruction and image streams from the final dual-stream layer—which have already been deeply comprehended and modeled—into a unified representation, and then it performs comprehensive multi-modal fusion and analysis, processing the sequence layer by layer to yield the final output. The computational logic within this layer is unified and modality-agnostic; both instruction and image inputs share the same parameters, including the QKV projections. As in the dual-stream layer, the bold lines denote the main path, while the thin lines represent the branch paths. Retaining only the main path reduces the layer to an identity mapping. The branch paths are responsible for complex computations and merge into the main path at appropriate junctures.

B.4 Summary of Architectural Design

In this section, we summarize our architectural design and detail the specific hyperparameter choices. Specifically, our DiT architecture comprises $\mathcal{N}_s = 32$ single-stream layers and $\mathcal{N}_d = 8$ dual-stream layers. We incorporate 3D Rotary Positional Embeddings (RoPE) (following [121, 131]) and, by default, utilize the hidden states from the final layer of the Instruction Encoder as the condition for the DiT. The base DiT model contains 10,292.56M (**~10B**) trainable parameters. When the optional Prompt Tuning Transformer is included, the total trainable parameter count slightly increases to 11,022.55M. Further architectural details and hyperparameter settings are comprehensively summarized in Table 10.

Parameter	Value	Parameter	Value
<i>Base Architecture</i>			
Patch Size	2	Single-Stream Layers	32
Input Channels	16	Dual-Stream Layers	8
Hidden Size	3360	Refiner Layers	2
Attention Heads	28	KV Heads	7
FFN Multiple Of	256	RoPE Axes Dims	[40, 40, 40]
<i>Instruction Feature Configuration</i>			
Feature Layers	1	Feature Dimension	4096
Reduction Type	Mean		
<i>Pretrained Modules</i>			
VAE Model	FLUX.1-dev [7]	Instruction Encoder	Qwen3-VL-8B-Instruct [3]
<i>Prompt Tuning (Optional)</i>			
Trainable Tokens	32	Causal Mask	True
Hidden Size	4096	Layers	3
Attention Heads	32	KV Heads	8
<i>Trainable Parameters</i>			
DiT Model	10,292.56M	Total (w/ Prompt Tuning)	11,022.55M

Table 10 Summary of architectural design, specific hyperparameter configurations, and parameter counts.

B.5 Training Process

Our overall training pipeline follows a progressive curriculum: we first train on low-resolution images before scaling to higher resolutions, and initially train the text-to-image (T2I) model before fine-tuning it into the text-and-image-to-image (TI2I) editing model. The complete training process is summarized in Table 11.

The parameters *Max Output Pixels*, *Max Input Pixels*, and *Max Side Length* define the spatial constraints during training. During data preprocessing, images exceeding these thresholds are dynamically downscaled to fit within the limits, while smaller or equal-sized images remain unchanged, inherently enabling mixed-resolution training.

For the TI2I training phase, we construct a mixed dataset of approximately 22M samples. This comprises 11.29M pure T2I samples sampled from the 47.19M Boogu Syllabus dataset (see 3.2.2), and 10.70M editing samples augmented from our initially constructed 2.50M TI2I dataset. Consequently, our image editing model is optimized through a combination of mixed-resolution and mixed-task training.

Parameter	T2I			TI2I
	512×512	1024×1024	2048×2048	2048×2048
Resolution & Data Constraints				
Training Data	Open-Source 187M	Open-Source 60M + Boogu Syllabus 47M	Boogu Syllabus 47M	11M T2I + 11M TI2I
Max Output Pixels	262,144	1,048,576	4,194,304	4,194,304
Max Input Pixels	262,144	1,048,576		
Max Side Length	800	2048		
Max Instruction Tokens	256	512		
Instruct. Dropout Prob.	0.1			944
Ref. Img. Dropout Prob.	N/A			
Batch Size & Steps				
Global Batch Size	1280	2048	1024	448
Micro Batch Size	10	8	2	1
Num Train Epochs	1	1	2	
Gradient Accumulation				
Optimizer & Learning Rate				
Learning Rate	1e-4			3e-5
Warmup Steps	1200	Constant with Warmup [47, 128, 129]	800	
LR Scheduler				
Warmup Init LR		1e-6		
Adam β_1, β_2		0.9, 0.95		
Adam Weight Decay		0.01		
Max Grad Norm		1.0		
Transport & Loss				
SNR Strategy	Dynamic Logit Normal [7, 36, 79, 84]		Rectified Dynamic Logit Normal (3.2.3)	
Dynamic Time Shifting	True			
Hardware & Precision				
Sharding Strategy	HYBRID_SHARD_ZERO2 [103, 144]		HYBRID_SHARD [48, 144]	
Mixed Precision	bf16			
Gradient Checkpointing	True			

Table 11 Training hyperparameters across different tasks and resolutions

B.6 Lightweight Module Details

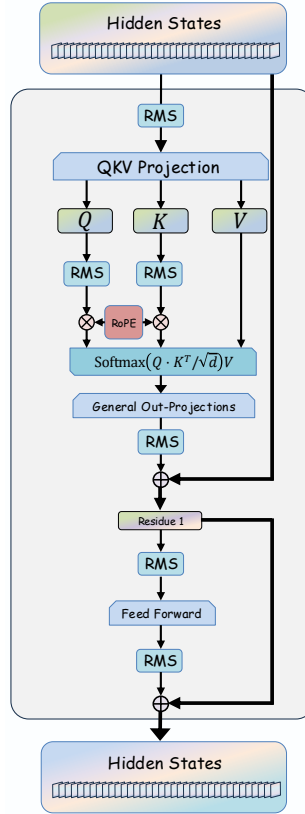


Figure 37 Lightweight single-stream transformer layer. This is a lightweight transformer module designed for various refiners and optional prompt tuning components. This module adopts the classic design [24, 29, 36, 38, 79, 97, 120, 127, 130–132, 147] with minimal modifications.

C Boosted Orthogonal Guidance

Classifier-free guidance [55] and related inference-time steering techniques have become indispensable for improving prompt adherence in text-to-image DiTs. However, aggressively amplifying guidance often trades alignment for degraded appearance, manifesting as oversaturation, flattened tones, over-smoothing, and visually implausible artifacts. In this section, we introduce **Boosted Orthogonal Guidance** (BOG), a new training-free inference method for DiT-based text-to-image generation that targets this failure mode while explicitly enhancing fine-grained details and cinematic atmosphere.

Inspired by the key insight behind APG [109] – namely that guidance updates can be meaningfully decomposed into components parallel and orthogonal to the model’s predicted flow – BOG emphasizes the information-preserving orthogonal contribution to avoid color blow-up and texture collapse. Crucially, BOG departs from the conventional view that a DiT prediction is merely a flattened high-dimensional vector: we treat each per-step predicted tensor as a 2D matrix aligned with the image’s inherent spatial structure, echoing the matrix-aware perspective adopted in optimizer design (e.g., Moun-style [63] updates). Building on this view, BOG applies an efficient orthogonalization approximation to the predicted matrix at every denoising step, acting as a structured normalization that not only controls magnitude but also enriches the diversity of row/column directions. Unlike vector-norm rescaling – which preserves a single global direction while adjusting only scale – our matrix-structured normalization encourages higher effective rank, thereby retaining more informative variation across spatial dimensions. Empirically, this rank-preserving, orthogonality-boosted guidance leads to richer micro-structures, improved tonal depth, and more coherent “film-like” lighting and ambience, while mitigating oversaturation without introducing additional training or meaningful sampling overhead.

C.1 Matrix-Structured View of DiT Predictions

Standard inference in DiTs typically treats the model prediction at each flow-matching (or denoising) step as a single flattened 1D update direction in a high-dimensional state space: under conditional or unconditional prompting, the current latent state (often a noisy or partially denoised image representation) is moved along this vector field toward a corresponding clean-image hidden state [56, 79, 84–86, 115, 117, 119]. While this viewpoint is mathematically convenient, it implicitly discards a key inductive bias of the image modality – namely, that the predicted hidden state is inherently spatial. Concretely, the DiT output is a tensor of shape $[B, C, H, W]$; after ignoring batch and channel dimensions, the remaining $[H, W]$ structure is not an arbitrary reshaping artifact but a meaningful 2D grid aligned with the image plane. This mismatch becomes more pronounced in how normalization is commonly applied: prevailing “norm” or rescaling strategies first collapse the 2D prediction into a 1D vector and then rescale its Euclidean length, which changes only the magnitude in the vector field while leaving the update direction (and its spatial composition) essentially untouched. Others additionally subtract a mean term – either the batch mean or the per-sample mean over features – thereby performing mean-centering. As a result, such vector-level normalization still overlooks spatial organization that could be leveraged to preserve informative variation and avoid overly smooth updates. Motivated by recent optimizer designs that explicitly respect the matrix/tensor structure rather than reducing updates to flat vectors [50, 63], we instead adopt a 2D normalization strategy that operates directly on the matrix-structured prediction at each step.

C.1.1 Matrix Normalization

Let $D \in \mathbb{R}^{m \times n}$ (Assume $m \geq n$, *w.l.o.g.*), we perform the singular value decomposition (SVD) of the matrix D as follows.

$$D = U \Sigma V^\top = U \cdot \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \\ 0 & 0 & \cdots & 0 \\ & & & \vdots \end{pmatrix} \cdot V^\top, \quad \Sigma \in \mathbb{R}^{m \times n}, \quad \text{w.l.o.g. } m \geq n, \quad (1)$$

We propose the following scalar function, whose inputs are the singular value index i and the singular value itself.

$$\sigma'_i = \text{norm}(\sigma_i, i) = \begin{cases} \text{Sigmoid}(\sigma_i), \\ \lambda \cdot \frac{\exp(\sigma_i/\tau)}{\sum_{j=1}^n \exp(\sigma_j/\tau)}, \\ \dots, \\ 1. \end{cases} \quad (2)$$

where the *norm* function can be chosen as needed; for example, we provide the options as above. Future work could further and more thoroughly investigate the differences, advantages, and disadvantages of these choices. Here λ is a hyperparameter, which can, for example, be set to around $(\frac{1}{n} \sum_{i=1}^n \delta_i)^{-1}$ in which $\delta_i = e^{\sigma_i/\tau} / \sum_{j=1}^n e^{\sigma_j/\tau}$. Thus, we propose the following matrix-oriented normalization method.

$$\text{MNorm}(\mathbf{D}) = \mathbf{U} \cdot \begin{pmatrix} \text{norm}(\sigma_1, 1) & 0 & \dots & 0 \\ 0 & \text{norm}(\sigma_2, 2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \text{norm}(\sigma_n, n) \\ \hline 0 & 0 & \dots & 0 \\ & \vdots & & \end{pmatrix} \cdot \mathbf{V}^\top \quad (3)$$

It is worth noting that when we set $\text{norm}(\sigma_i, i) = 1$, we obtain the following equivalent form.

$$\text{MNorm}(\mathbf{D}) = \arg \min_{\mathbf{O} \in \mathbb{R}^{m \times n}} \{ \|\mathbf{O} - \mathbf{D}\|_F \mid \mathbf{O}^\top \mathbf{O} = \mathbf{I}_n \text{ or } \mathbf{O} \mathbf{O}^\top = \mathbf{I}_m \} \quad (4)$$

Therefore, in this case, $\text{MNorm}(\mathbf{D})$ is essentially the optimal orthogonal (semi-orthogonal) approximation of the matrix $\mathbf{D}$ (see Appendix C.4 for a detailed proof). In practical image generation, we do not necessarily need to compute the SVD explicitly; instead, we can estimate $\text{MNorm}(\mathbf{D})$ using the Newton–Schulz method [63]. The specific algorithmic procedure is also provided in Appendix C.5.

C.1.2 Matrix Orthogonal Decomposition

Preliminary - Ordinary Differential Equation (ODE) We briefly review key concepts in diffusion models and introduce the notation used in this paper. $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$ denotes a sample, and here we let $\mathbf{x}_t = \alpha(t)\mathbf{x} + \sigma(t)\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, which describes the forward noising process in diffusion models. Here $\sigma(t)$ with $t \in [0, 1]$ gives the noise schedule, i.e., the amount of noise at the time step t . Following [131, 147], we adopt the setting $\sigma(1)=0$ and $\sigma(0)=\sigma_{\max}$, so that $t=0$ corresponds to (approximately) pure noise and $t=1$ corresponds to a clean sample. Building upon the Probability Flow ODE (PF-ODE) framework [65, 119], the forward diffusion process can be equivalently described by a deterministic ODE that preserves the marginal probability densities. From this continuous-time perspective, the generative process simply amounts to solving this empirical ODE to map a pure noise distribution to the clean data manifold, which is formulated as:

$$d\mathbf{x}_t = -\sigma(t)\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)d\sigma(t) = -\dot{\sigma}(t)\sigma(t)\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)dt \quad \#^1 \quad (5)$$

where $p_t(\mathbf{x}_t)$ represents the marginal distribution of the perturbed data at time t , bridging the Gaussian noise $p_0 = \mathcal{N}(\mathbf{0}, \sigma_{\max}^2 \mathbf{I})$ and the target data distribution $p_1 = p_{\text{data}}$. Generative sampling is performed by numerically integrating this ODE forward from $t=0$ to $t=1$. To operationalize this process, the analytically intractable score $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ is approximated by a Diffusion Transformer (DiT) $\mathbf{D}_\theta(\mathbf{x}_t, t)$, which is optimized to recover the uncorrupted sample $\mathbf{x}$ from the noisy observation $\mathbf{x}_t$. For conditional generation, the DiT architecture is readily augmented to $\mathbf{D}_\theta(\mathbf{x}_t, t, \mathbf{c})$, enabling auxiliary inputs $\mathbf{c}$, such as text instructions or class labels, to guide the generation.

^{#1} For our setting $\sigma(1)=0$ and $\sigma(0)=\sigma_{\max}$, $\sigma(t) \geq 0$ is a monotonically decreasing function with $\dot{\sigma}(t) \leq 0$, and $dt \geq 0$ during generation.

Preliminary - Classifier-Free Guidance (CFG) CFG [55] is a widely adopted inference technique that enhances sample quality and condition alignment by extrapolating between conditional and unconditional model predictions. Given a null condition⁰ $\mathbf{c}_\emptyset$, CFG modifies the output of the DiT denoiser at each sampling time step t as follows:

$$\begin{aligned} \mathbf{D}_{CFG}(\mathbf{x}_t, t, \mathbf{c}) &= \mathbf{D}_\theta(\mathbf{x}_t, t, \mathbf{c}_\emptyset) + \omega (\mathbf{D}_\theta(\mathbf{x}_t, t, \mathbf{c}) - \mathbf{D}_\theta(\mathbf{x}_t, t, \mathbf{c}_\emptyset)) \\ &= \mathbf{D}_\theta(\mathbf{x}_t, t, \mathbf{c}) + (\omega - 1) \Delta \mathbf{D}_{\theta, \mathbf{x}_t, t} \end{aligned} \quad (6)$$

where $\Delta \mathbf{D}_{\theta, \mathbf{x}_t, t} = \mathbf{D}_\theta(\mathbf{x}_t, t, \mathbf{c}) - \mathbf{D}_\theta(\mathbf{x}_t, t, \mathbf{c}_\emptyset)$. The DiT output is typically a tensor of shape $[B, C, H, W]$; in some cases for simplicity, we can omit the batch size B and channel dimension C , treating it as an $[H, W]$ matrix. The unconditional prediction $\mathbf{D}_\theta(\mathbf{x}_t, t, \mathbf{c}_\emptyset)$ is conventionally obtained by randomly dropping the condition $\mathbf{c}$ during training, though a separate unconditional DiT denoiser can also be trained for this purpose [66]. Similar to the truncation trick in GANs [10], CFG improves image quality at the expense of generation diversity [88]. While a larger guidance scale ω generally yields better instruction alignment, recent studies indicate that at high ω values, the parallel component of $\Delta \mathbf{D}_{\theta, \mathbf{x}_t, t}$ with respect to $\mathbf{D}_\theta(\mathbf{x}_t, t, \mathbf{c})$ (when flattened into a 1D vector) leads to image over-saturation [109].

Parallel-Orthogonal Decomposition For notational convenience, we denote $\Delta \mathbf{D} := \Delta \mathbf{D}_{\theta, \mathbf{x}_t, t} \in \mathbb{R}^{H \times W}$ and $\mathbf{D}^{(c)} := \mathbf{D}_\theta(\mathbf{x}_t, t, \mathbf{c}) \in \mathbb{R}^{H \times W}$. Furthermore, let $\mathbf{D}_{[i,:]}^{(c)}$ denote the i -th row vector of the DiT output matrix $\mathbf{D}^{(c)}$, and similarly, let $\mathbf{D}_{[:,j]}^{(c)}$ denote its j -th column vector; this notation applies analogously to all other matrices. For instance, taking a column vector $\mathbf{D}_{[:,j]}^{(c)}$ from the conditional DiT output matrix as the reference direction, we can perform a parallel-orthogonal decomposition using vector functions $\text{Parall}(\cdot, \cdot)$ and $\text{Ortho}(\cdot, \cdot)$. Akin to Gram-Schmidt orthogonalization, this process is formulated as:

$$\begin{aligned} \Delta \mathbf{D}_{[:,j]}^\parallel &= \text{Parall}(\mathbf{D}_{[:,j]}, \mathbf{D}_{[:,j]}^{(c)}) = \frac{\Delta \mathbf{D}_{[:,j]}^\top \mathbf{D}_{[:,j]}^{(c)}}{\|\mathbf{D}_{[:,j]}^{(c)}\|_2^2} \cdot \mathbf{D}_{[:,j]}^{(c)} \\ \Delta \mathbf{D}_{[:,j]}^\perp &= \text{Ortho}(\mathbf{D}_{[:,j]}, \mathbf{D}_{[:,j]}^{(c)}) = \Delta \mathbf{D}_{[:,j]} - \Delta \mathbf{D}_{[:,j]}^\parallel \end{aligned} \quad (7)$$

By applying the aforementioned decomposition to each column j , we obtain the full matrices $\Delta \mathbf{D}_{\text{col}}^\parallel \in \mathbb{R}^{H \times W}$ and $\Delta \mathbf{D}_{\text{col}}^\perp \in \mathbb{R}^{H \times W}$. Similarly, performing this decomposition on each row i yields the matrices $\Delta \mathbf{D}_{\text{row}}^\parallel, \Delta \mathbf{D}_{\text{row}}^\perp \in \mathbb{R}^{H \times W}$. This process is formulated as:

$$\begin{aligned} \Delta \mathbf{D}_{\text{col}}^\parallel &= \text{Stack} \left(\left\{ \text{Parall}(\mathbf{D}_{[:,j]}, \mathbf{D}_{[:,j]}^{(c)}) \mid j = 0, \dots, W-1 \right\} \right) \\ \Delta \mathbf{D}_{\text{row}}^\parallel &= \text{Stack} \left(\left\{ \text{Parall}(\mathbf{D}_{[i,:]}, \mathbf{D}_{[i,:]}^{(c)}) \mid i = 0, \dots, H-1 \right\} \right) \end{aligned} \quad (8)$$

Similarly, $\Delta \mathbf{D}_{\text{col}}^\perp$ and $\Delta \mathbf{D}_{\text{row}}^\perp$ can be obtained using $\text{Ortho}(\cdot, \cdot)$. In summary, we have:

$$\Delta \mathbf{D}_{\text{pod}} = \text{POD}(\Delta \mathbf{D}, \mathbf{D}^{(c)}) = \lambda_c \cdot (\Delta \mathbf{D}_{\text{col}}^\perp + \mu \Delta \mathbf{D}_{\text{col}}^\parallel) + \lambda_r \cdot (\Delta \mathbf{D}_{\text{row}}^\perp + \mu \Delta \mathbf{D}_{\text{row}}^\parallel) \quad (9)$$

where $\lambda_c = \frac{W}{H+W+\zeta}$ and $\lambda_r = \frac{H}{H+W+\zeta}$, with ζ being a small constant. The hyperparameter μ controls the magnitude of the parallel components. In practice, we set $\mu \leq 1$ (e.g., 0, 0.1, 0.25) to mitigate image over-saturation when ω is relatively large [109].

C.2 Rolling-Sum Momentum

Inspired by momentum strategies commonly used in optimization, we introduce a similar mechanism during the inference of text-to-image generation. This is motivated by the intuition that the "inertia" of the trajectory within the flow matching vector field influences the final generation. Our momentum update is simply

⁰Here, $\mathbf{c}_{\text{neg}}$ denotes a context condition derived from negative text prompts (e.g., "low quality, deformed, blurry, ..."). In practice, many models substitute the standard empty condition $\mathbf{c}_\emptyset$ with $\mathbf{c}_{\text{neg}}$. *w.l.o.g.*, we use $\mathbf{c}_\emptyset$ to broadly represent either the true unconditional state $\mathbf{c}_\emptyset$ or $\mathbf{c}_{\text{neg}}$.

formulated as $\overline{\Delta D}_{(i)} = \eta \Delta D_{(i)} + \rho \overline{\Delta D}_{(i-1)}$, with typical default values of $\eta = 0.9$ and $\rho = 0.1$. Here, $i \in \mathbb{N}_+$ denotes the inference iteration step, which is related to the diffusion time step $t \in \mathbb{R}$ fed into the DiT via a scheduler function, i.e., $t = \text{scheduler}(i)$. We abstract this momentum update as a function:

$$\overline{\Delta D}_{(i)} = \text{RollSumUpdate}(\Delta D, i) = \eta \Delta D_{(i)} + \rho \overline{\Delta D}_{(i-1)} \quad \Big| \quad \overline{\Delta D}_{(0)} = \mathbf{0} \quad (10)$$

This contrasts with recent findings by [109], who propose a "reverse momentum" (i.e., $\rho < 0$). They argue that this pushes the model away from previous CFG update directions and encourages it to focus more on the current one. However, our empirical observations indicate that setting $\rho < 0$ causes directional jitter for $\overline{\Delta D}_{(i)}$ within the vector field across iterations.

C.3 Putting It All Together

We now present the complete pipeline of our proposed method. Algorithm 1 summarizes the computation of the Boosted Orthogonal Guidance (BOG) at a single inference iteration step i .

By explicitly treating the predicted tensors as spatial matrices, BOG systematically refines the guidance direction. It first smooths the trajectory using momentum, then normalizes the matrix to preserve spatial rank, and finally isolates the orthogonal components—effectively diminishing the projections parallel to the conditional prediction—to prevent oversaturation. Ultimately, this rank-preserving and orthogonality-boosted approach significantly elevates the visual quality, producing richer micro-details, enhanced tonal depth, and a realistic cinematic lighting, all achieved zero-shot without meaningful overhead.

Algorithm 1 Single Step of Boosted Orthogonal Guidance (BOG)

Input: Inference step $i \in \mathbb{N}_+$, current latent state $\mathbf{x}_t$, condition $\mathbf{c}$, unconditional state $\mathbf{c}_\emptyset$, guidance scale ω .

Output: The BOG modified prediction $D_{BOG(i)}$.

- | | |
|---|---|
| 1: $t \leftarrow \text{scheduler}(i)$ | ▷ Obtain the continuous diffusion time step |
| 2: $\Delta D_{(i)}^{(c)} \leftarrow D_\theta(\mathbf{x}_t, t, \mathbf{c}), \quad \Delta D_{(i)}^{(c_\emptyset)} \leftarrow D_\theta(\mathbf{x}_t, t, \mathbf{c}_\emptyset)$ | ▷ Compute DiT predictions |
| 3: $\Delta D_{(i)} \leftarrow \Delta D_{(i)}^{(c)} - \Delta D_{(i)}^{(c_\emptyset)}$ | ▷ Calculate the raw guidance direction |
| 4: $\widehat{\Delta D}_{(i)} \leftarrow \text{MNorm}\left(\text{RollSumUpdate}(\Delta D_{(i)}, i)\right)$ | ▷ Apply momentum & matrix normalization |
| 5: $\widehat{\Delta D}_{(i)} \leftarrow \text{POD}(\widehat{\Delta D}_{(i)}, \Delta D_{(i)}^{(c)})$ | ▷ Parallel-Orthogonal Decomposition |
| 6: $D_{BOG(i)} \leftarrow \Delta D_{(i)}^{(c)} + (\omega - 1) \widehat{\Delta D}_{(i)}$ | ▷ Compute final BOG tensor |
| 7: return $D_{BOG(i)}$ | |
-

Limitations While BOG effectively enhances the cinematic atmosphere of generated images, it introduces certain limitations. Specifically, it can increase the probability of structural distortions and reduce the success rate of text rendering. Consequently, we treat BOG as an optional module, akin to a selectable filter.

To mitigate these issues, we introduce a simple hyperparameter: the BOG Interval (Δ_{BOG}). During inference, BOG is applied every Δ_{BOG} diffusion steps, while standard CFG is used for the remaining steps. For instance, in a 10-step denoising process with $\Delta_{\text{BOG}} = 2$, BOG is executed on even steps (2, 4, 6, 8, 10) and standard CFG on odd steps. Thus, $\Delta_{\text{BOG}} = 1$ denotes full BOG inference. We recommend $\Delta_{\text{BOG}} = 2$ as the default setting, which significantly alleviates the aforementioned artifacts, albeit with a slight trade-off in the enhanced BOG-styled photographic texture.

C.4 Optimal Semi-Orthogonal Approximation

Proof. Let $\mathbf{D} \in \mathbb{R}^{m \times n}$ with $m \geq n$, and let $\mathbf{O} \in \mathbb{R}^{m \times n}$ be semi-orthogonal, i.e. $\mathbf{O}^\top \mathbf{O} = \mathbf{I}_n$. Let the (thin) SVD of $\mathbf{D}$ be

$$\mathbf{D} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top,$$

where $\mathbf{U} \in \mathbb{R}^{m \times n}$ satisfies $\mathbf{U}^\top \mathbf{U} = \mathbf{I}_n$, $\mathbf{V} \in \mathbb{R}^{n \times n}$ is orthogonal, and

$$\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_n), \quad \sigma_1 \geq \dots \geq \sigma_n \geq 0.$$

Remark on $\mathbf{Q}$. Although $\mathbf{U}$ and $\mathbf{O}$ are rectangular (semi-orthogonal), the matrix

$$\mathbf{Q} := \mathbf{U}^\top \mathbf{O} \mathbf{V}$$

is square ($n \times n$). Moreover,

$$\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_n,$$

so $\mathbf{Q}$ is an orthogonal matrix in $\mathbb{R}^{n \times n}$ (not merely semi-orthogonal). Indeed, for a square matrix, $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_n$ already implies $\mathbf{Q}$ is orthogonal (equivalently, $\mathbf{Q}^{-1} = \mathbf{Q}^\top$ and also $\mathbf{Q} \mathbf{Q}^\top = \mathbf{I}_n$).

Step 1 (Reduction to trace maximization). Expand the squared Frobenius norm:

$$\begin{aligned} \|\mathbf{D} - \mathbf{O}\|_F^2 &= \text{tr}((\mathbf{D} - \mathbf{O})^\top (\mathbf{D} - \mathbf{O})) \\ &= \text{tr}(\mathbf{D}^\top \mathbf{D}) + \text{tr}(\mathbf{O}^\top \mathbf{O}) - 2 \text{tr}(\mathbf{O}^\top \mathbf{D}) \\ &= \|\mathbf{D}\|_F^2 + \|\mathbf{O}\|_F^2 - 2 \text{tr}(\mathbf{O}^\top \mathbf{D}). \end{aligned}$$

Since $\mathbf{O}^\top \mathbf{O} = \mathbf{I}_n$,

$$\|\mathbf{O}\|_F^2 = \text{tr}(\mathbf{O}^\top \mathbf{O}) = \text{tr}(\mathbf{I}_n) = n.$$

Thus minimizing $\|\mathbf{D} - \mathbf{O}\|_F^2$ over $\mathbf{O}^\top \mathbf{O} = \mathbf{I}_n$ is equivalent to maximizing $\text{tr}(\mathbf{O}^\top \mathbf{D})$.

Step 2 (SVD coordinates). Using $\mathbf{D} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top$ and cyclicity of the trace,

$$\begin{aligned} \text{tr}(\mathbf{O}^\top \mathbf{D}) &= \text{tr}(\mathbf{O}^\top \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top) \\ &= \text{tr}(\mathbf{V}^\top \mathbf{O}^\top \mathbf{U} \mathbf{\Sigma}) \\ &= \text{tr}(\mathbf{Q}^\top \mathbf{\Sigma}), \end{aligned}$$

where $\mathbf{Q} = \mathbf{U}^\top \mathbf{O} \mathbf{V} \in \mathbb{R}^{n \times n}$.

Step 3 (Elementwise bound via diagonal entries). Since $\mathbf{\Sigma}$ is diagonal,

$$\text{tr}(\mathbf{Q}^\top \mathbf{\Sigma}) = \sum_{i=1}^n \sigma_i q_{ii}.$$

For each i , let $\mathbf{e}_i$ be the i th standard basis vector. Then $q_{ii} = \mathbf{e}_i^\top \mathbf{Q} \mathbf{e}_i$ and

$$\begin{aligned} |q_{ii}| &= |\mathbf{e}_i^\top \mathbf{Q} \mathbf{e}_i| \leq \|\mathbf{e}_i\|_2 \|\mathbf{Q} \mathbf{e}_i\|_2 \\ &\leq \|\mathbf{e}_i\|_2 \|\mathbf{Q}\|_2 \|\mathbf{e}_i\|_2. \end{aligned}$$

Because $\mathbf{Q}$ is orthogonal, $\|\mathbf{Q}\|_2 = 1$; also $\|\mathbf{e}_i\|_2 = 1$. Hence $|q_{ii}| \leq 1$. Since $\sigma_i \geq 0$, we obtain

$$\text{tr}(\mathbf{Q}^\top \mathbf{\Sigma}) = \sum_{i=1}^n \sigma_i q_{ii} \leq \sum_{i=1}^n \sigma_i |q_{ii}| \leq \sum_{i=1}^n \sigma_i.$$

Therefore, for all feasible $\mathbf{O}$,

$$\text{tr}(\mathbf{O}^\top \mathbf{D}) \leq \sum_{i=1}^n \sigma_i.$$

Step 4 (Achievability). Let

$$\mathbf{O}_\star := \mathbf{U}\mathbf{V}^\top.$$

Then $\mathbf{O}_\star^\top \mathbf{O}_\star = \mathbf{V}\mathbf{U}^\top \mathbf{U}\mathbf{V}^\top = \mathbf{I}_n$, and

$$\mathbf{Q}_\star = \mathbf{U}^\top \mathbf{O}_\star \mathbf{V} = \mathbf{U}^\top (\mathbf{U}\mathbf{V}^\top) \mathbf{V} = \mathbf{I}_n.$$

Hence

$$\text{tr}(\mathbf{O}_\star^\top \mathbf{D}) = \text{tr}(\mathbf{Q}_\star^\top \mathbf{\Sigma}) = \text{tr}(\mathbf{\Sigma}) = \sum_{i=1}^n \sigma_i,$$

so the upper bound is tight. Thus $\mathbf{O}_\star$ maximizes $\text{tr}(\mathbf{O}^\top \mathbf{D})$ and therefore minimizes $\|\mathbf{D} - \mathbf{O}\|_F$.

Minimum value. Using Step 1 and $\max_{\mathbf{O}^\top \mathbf{O} = \mathbf{I}_n} \text{tr}(\mathbf{O}^\top \mathbf{D}) = \sum_{i=1}^n \sigma_i$,

$$\min_{\mathbf{O}^\top \mathbf{O} = \mathbf{I}_n} \|\mathbf{D} - \mathbf{O}\|_F^2 = \|\mathbf{D}\|_F^2 + n - 2 \sum_{i=1}^n \sigma_i.$$

□

Supplement (same bound via Von Neumann's trace inequality). With $\mathbf{Q} = \mathbf{U}^\top \mathbf{O} \mathbf{V}$ we have $\text{tr}(\mathbf{O}^\top \mathbf{D}) = \text{tr}(\mathbf{Q}^\top \mathbf{\Sigma})$. Von Neumann's trace inequality [89] states that for any $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$,

$$|\text{tr}(\mathbf{A}^\top \mathbf{B})| \leq \sum_{i=1}^n \sigma_i(\mathbf{A}) \sigma_i(\mathbf{B}).$$

Taking $\mathbf{A} = \mathbf{Q}$ and $\mathbf{B} = \mathbf{\Sigma}$ gives

$$\text{tr}(\mathbf{Q}^\top \mathbf{\Sigma}) \leq |\text{tr}(\mathbf{Q}^\top \mathbf{\Sigma})| \leq \sum_{i=1}^n \sigma_i(\mathbf{Q}) \sigma_i(\mathbf{\Sigma}).$$

Since $\mathbf{Q}$ is orthogonal, $\sigma_i(\mathbf{Q}) = 1$ for all i , and $\sigma_i(\mathbf{\Sigma}) = \sigma_i$, hence

$$\text{tr}(\mathbf{O}^\top \mathbf{D}) = \text{tr}(\mathbf{Q}^\top \mathbf{\Sigma}) \leq \sum_{i=1}^n \sigma_i,$$

with equality at $\mathbf{Q} = \mathbf{I}_n$ (equivalently, $\mathbf{O} = \mathbf{U}\mathbf{V}^\top$), recovering the same optimizer and the same minimum value.

C.5 Newton-Schulz Approximation

To compute $\text{MNorm}(\mathbf{D})$, it is not strictly necessary to perform Singular Value Decomposition (SVD). Specifically, for the case where $\text{norm}(\sigma_i, i) = 1$, we can directly estimate it using the Newton-Schulz method, following the approach of [63] (see Code 1).

```

1 def _newtonschulz5_batched(G: torch.Tensor, steps: int = 5, eps: float = 1e-7):
2     """
3     Batched Newton-Schulz iteration.
4
5     Accepts:
6     - (H, W)           -> returns (H, W)
7     - (N, H, W)        -> returns (N, H, W)
8     - (B, C, H, W)     -> returns (B, C, H, W)
9     """
10    a, b, c = (3.4445, -4.7750, 2.0315)
11
12    orig_ndim = G.ndim
13    if orig_ndim == 2:
14        G3 = G.unsqueeze(0)           # (1, H, W)
15        out_shape = None
16    elif orig_ndim == 3:
17        G3 = G                       # (N, H, W)
18        out_shape = None
19    elif orig_ndim == 4:
20        B, C, H, W = G.shape
21        G3 = G.reshape(B * C, H, W)  # (N, H, W)
22        out_shape = (B, C, H, W)
23    else:
24        raise ValueError(f"Expected 2D/3D/4D tensor, got ndim={G.ndim}")
25
26    # Match the original behavior: decide whether to transpose based on H/W
27    H, W = G3.shape[-2], G3.shape[-1]
28
29    # Compute in bfloat16 (keeps the original logic)
30    X = G3.to(torch.bfloat16)
31
32    # Normalize each matrix by its Frobenius norm: X /= (||X||_F + eps)
33    # Frobenius norm = sqrt(sum_ij X^2)
34    nrm = torch.linalg.norm(X, ord="fro", dim=(-2, -1))  # (N,)
35    X = X / (nrm.unsqueeze(-1).unsqueeze(-1) + eps)
36
37    transposed = False
38    if H > W:
39        # Transpose the last two dims so we iterate on the "shorter" dimension first
40        X = X.transpose(-2, -1)  # (N, W, H)
41        transposed = True
42
43    # Newton-Schulz iterations (batched GEMMs)
44    for _ in range(steps):
45        A = X @ X.transpose(-2, -1)  # (N, m, m)
46        Bm = b * A + c * (A @ A)    # (N, m, m)
47        X = a * X + (Bm @ X)        # (N, m, n)
48
49    # Transpose back if we transposed at the beginning
50    if transposed:
51        X = X.transpose(-2, -1)
52
53    # Restore original shape
54    if orig_ndim == 2:
55        return X.squeeze(0)
56    if out_shape is not None:
57        return X.reshape(out_shape)
58    return X

```

Code 1 Batched Newton-Schulz Iteration in PyTorch