Gen-Searcher: Reinforcing Agentic Search for Image Generation

Kaituo Feng[1] **Manyuan Zhang**[1†] **Shawn Chen**[2] **Yunlong Lin**[1] **Kaixuan Fan**[1]
Yilei Jiang[1] **Hongyu Li** **Dian Zheng**[1] **Chenyang Wang**[3] **Xiangyu Yue**[1†]
[1]MMLab, CUHK [2]UCLA [3]UC Berkeley

Home: https://gen-searcher.vercel.app/



Figure 1: Generated images using our proposed Gen-Searcher.

Abstract

Recent image generation models have shown strong capabilities in generating high-fidelity and photorealistic images. However, they are fundamentally constrained by frozen internal knowledge, thus often failing on real-world scenarios that are knowledge-intensive or require up-to-date information. In this paper, we present Gen-Searcher, as the first attempt to train a search-augmented image generation agent, which performs multi-hop reasoning and search to collect the textual knowledge and reference images needed for grounded generation. To achieve this, we construct a tailored data pipeline and curate two high-quality datasets, Gen-Searcher-SFT-10k and Gen-Searcher-RL-6k, containing diverse search-intensive prompts and corresponding ground-truth synthesis images. We further introduce KnowGen, a comprehensive benchmark that explicitly requires search-grounded external knowledge for image generation and evaluates models from multiple dimensions. Based on these resources, we train Gen-Searcher with SFT followed by agentic reinforcement learning with dual reward feedback, which combines text-based and image-based rewards to provide more stable and informative learning signals for GRPO training. Experiments show that Gen-Searcher brings substantial

†Corresponding Authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

gains, improving Qwen-Image by around 16 points on KnowGen and 15 points on WISE. We hope this work can serve as an open foundation for search agents in image generation, and we fully open-source our data, models, and code.

1 Introduction

Recent text-to-image generation models have achieved remarkable progress in producing high-fidelity and photorealistic images [1, 2, 3]. Despite these advances, most of them remain fundamentally limited by frozen internal knowledge acquired during pretraining. As a result, these models often struggle with real-world prompts that are knowledge-intensive or require up-to-date information. For instance, generating images involving specific landmarks, public figures, newly released products, or other evolving real-world entities often requires external knowledge that cannot be reliably inferred from the model's parametric memory alone. Currently, only a few advanced proprietary models like Nano Banana Pro [2], support search before generation, yet they remain limited to text search without retrieving visual references.

To mitigate this limitation, prior work has explored RAG-based approaches that retrieve relevant knowledge from external databases to support generation [4, 5, 6]. However, these methods are limited by the coverage and freshness of static databases, which cannot fully capture the vast and evolving knowledge of the real world. In addition, the similarity-based single-round shallow retrieval makes them inadequate for complex real-world queries that require deep search. Besides, a few prompt-based workflows [7, 8] are recently proposed to enhance image generation models by directly searching for information on the web. However, these works rely on manually designed prompting strategies to guide search and generation without training. As a result, the search behavior is often brittle and suboptimal, lacking the ability to adaptively plan search steps, refine queries, or reason over retrieved evidence. Inspired by the recent success of agentic reinforcement learning (agentic RL) [9, 10] in deep research tasks, we ask a natural question: can we train a search agent for image generation that actively performs multi-hop web search and reasoning to gather knowledge in the web for grounded image generation?

In this paper, we introduce Gen-Searcher, as the first attempt to train a multimodal deep search agent for image generation. To achieve this goal, we build a dedicated data pipeline, in order to overcome the lack of suitable training data for this task. Specifically, we construct search-intensive image generation prompts from two sources: multi-hop prompts generated by Gemini 3 Pro [11] with carefully designed prompt engineering across around 20 diverse categories, and converted deep-research QA datasets mainly covering general news. We then use Gemini 3 Pro with search tools to generate agentic trajectories that gather textual evidence and reference images, which are further used to form search-grounded prompts. These prompts are fed into Nano Banana Pro to synthesize ground-truth images, followed by quality filtering based on Seed1.8 [12]. This pipeline yields two training datasets, Gen-Searcher-SFT-10k and Gen-Searcher-RL-6k, together with KnowGen, a benchmark with the proposed K-Score for evaluating real-world search-grounded image generation.

Based on the constructed training datasets, we train Gen-Searcher in two stages: SFT equips the model with basic tool-use abilities, enabling multi-step search, browsing, and reasoning for image generation; while agentic RL based on GRPO [13, 14] further optimizes its tool-calling trajectories, encouraging the model to produce higher-quality search-grounded prompts for generation. Notably, we find that this task poses unique challenges: due to the limited capability and high generation variance of open-source image generators such as Qwen-Image [1], sometimes even correct searched information may still fail to produce high-quality images, making pure image-based rewards noisy and unstable. To address this issue, we introduce an additional text-based reward that evaluates whether the collected search-grounded prompt contains sufficient information to generate the target image. We then combine both text-based and image-based rewards to train the agent with GRPO, providing more stable and informative feedback for optimizing the search and reasoning process.

We conduct extensive experiments to evaluate the effectiveness of Gen-Searcher. Our method consistently improves image generation performance across different backbones, bringing around 16-point gains for Qwen-Image on KnowGen. We further observe strong transferability: a Gen-Searcher trained with Qwen-Image can be directly applied to Seedream 4.5 [15] and Nano Banana Pro [2] without additional training, yielding about 16-point and 3-point improvements, respectively. Beyond our proposed KnowGen benchmark, we also evaluate on WISE [16], where Gen-Searcher improves

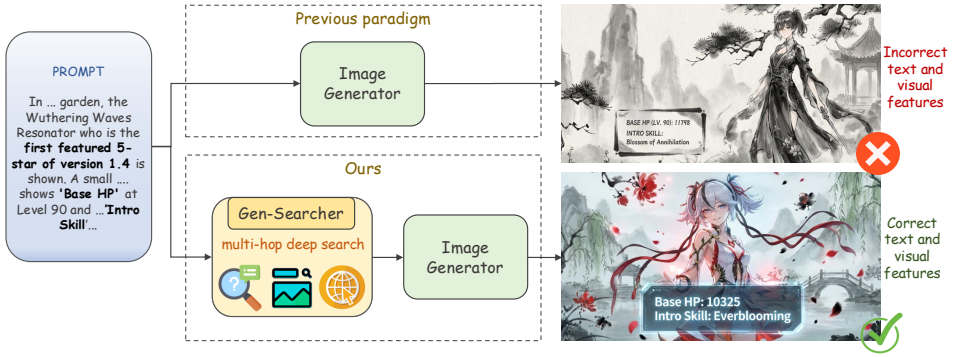


Figure 2: Our proposed Gen-Searcher enables search-grounded generation in real-world knowledge-intensive scenarios.

Qwen-Image from 0.62 to 0.77, demonstrating its strong generalization in knowledge-intensive image generation tasks. Our contributions can be summarized as follows:

- We propose **Gen-Searcher**, the first attempt to explore training a multimodal deep search agent for image generation. We fully open-source our project and hope that Gen-Searcher can serve as an open foundation for future research.
- To support training, we build a dedicated data pipeline to construct search-intensive image generation data, resulting in two training datasets, **Gen-Searcher-SFT-10k** and **Gen-Searcher-RL-6k**. In addition, we introduce **KnowGen**, a new benchmark designed to evaluate search-grounded image generation in knowledge-intensive real-world scenarios.
- Extensive experiments validate the effectiveness of our Gen-Searcher. For example, our method improves Qwen-Image by around 16 points on KnowGen and 15 points on WISE.

2 Method

2.1 Dataset Construction

High-quality training data is essential for developing a search agent capable of performing multi-hop deep search and reasoning for image generation. However, such data does not naturally exist, since it requires aligned pairs of search-intensive prompts, agentic search trajectories, and grounded images. To address this, we design a dedicated pipeline to construct training data, as illustrated in Figure 3.

Text Prompt Construction. We first construct text prompts that require deep web search before image generation. To ensure diversity, we adopt two complementary strategies. Our primary approach uses carefully designed prompt engineering to instruct **Gemini 3 Pro** [11] to generate multi-hop search-intensive prompts across a broad range of categories, including *Anime*, *Architecture*, *Art*, *Astronomy*, *Biology*, *Celebrities*, *Chemistry*, *Culture*, *Engineering*, *Film*, *Game*, *Geography*, *History*, *Industry*, *Medicine*, *Physics*, *Politics*, *Posters*, *Religion*, and *Sports*. As a complementary strategy, we convert samples in two existing search QA datasets [17, 9] into image-generation-oriented prompts. In particular, we use Gemini 3 Pro to transform questions into prompts that require generating a grounded visual depiction of the queried entity or event. This strategy primarily contributes prompts related to *General News*, further expanding the coverage of diverse knowledge scenarios.

Agentic Trajectory Generation. Given the constructed text prompts, we generate agentic search trajectories to perform deep search and gather sufficient evidence for producing the final search-grounded prompt together with selected reference images for accurate visual features. Meanwhile, these trajectories also serve as valuable supervision data for subsequent SFT. Specifically, we employ **Gemini 3 Pro** together with a set of search tools (introduced in Section 2.3) in a multi-turn manner. During this process, the agent continuously analyzes textual and visual feedback from the environment, identifies useful evidence and plans the next action accordingly. Through this multi-turn reasoning and search procedure, the agent progressively aggregates information from multiple sources before finally producing a grounded prompt and a set of relevant reference images for image synthesis.

Ground-Truth Image Synthesis. After obtaining the final grounded prompts and visual references, we synthesize the corresponding images using the proprietary image generation model **Nano Banana**

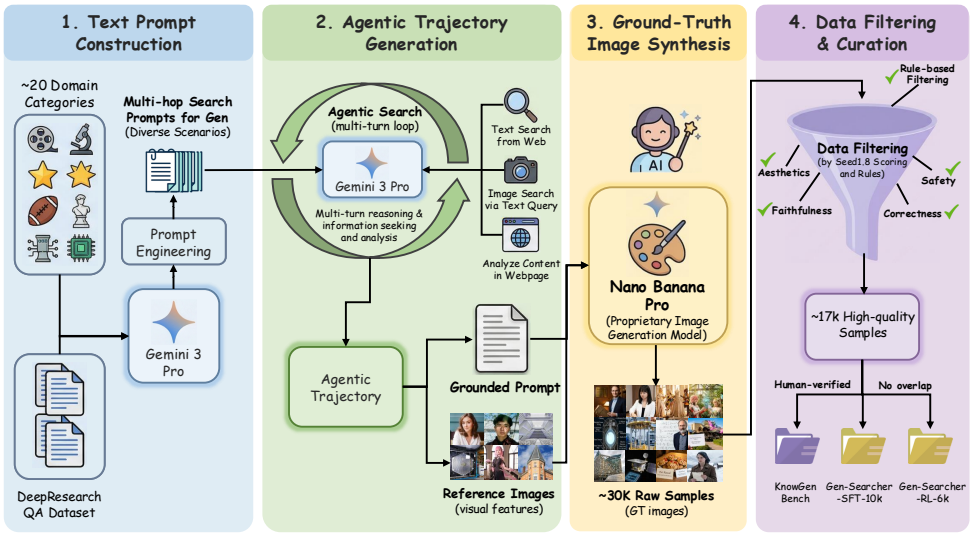


Figure 3: An illustration of our data curation pipeline.

Pro [2]. The generated images serve as synthesis ground truth for training the search agent. This process results in approximately **30K** raw samples consisting of prompts, search trajectories, grounded prompts, reference images, and ground-truth images.

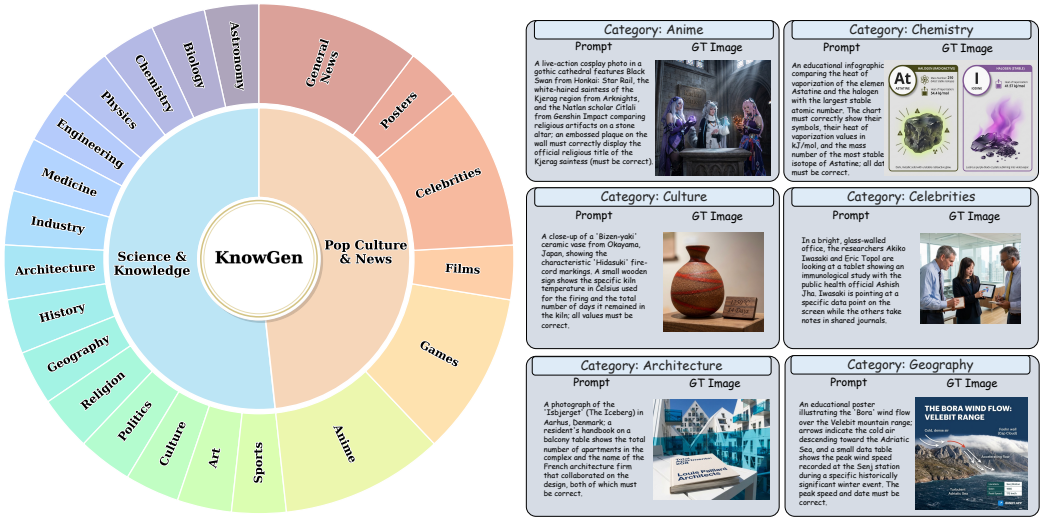
Data Filtering and Benchmark Construction. To ensure data quality, we further employ another strong proprietary model **Seed1.8** [12] to score the generated samples from multiple perspectives, including whether the prompt genuinely requires search, the correctness of the generated content, faithfulness to the prompt, visual aesthetics, text rendering clarity, and safety considerations. These model-based scores are combined with rule-based filtering, such as removing prompts with excessively long token lengths or inconsistent search results. After filtering, we obtain approximately **17K** high-quality samples. From this curated dataset, we select **630 human-verified samples** to construct a held-out benchmark named **KnowGen**. The remaining **16K** samples are used for training and are split into two datasets: **Gen-Searcher-SFT-10k** for SFT and **Gen-Searcher-RL-6k** for RL. We strictly ensure that no overlap exists between the training data and the evaluation benchmark.

2.2 KnowGen Benchmark

For evaluation, we introduce **KnowGen**, a comprehensive benchmark designed to evaluate image generation under real-world *knowledge-intensive* and *search-dependent* scenarios. Each sample in KnowGen is built to require non-trivial external knowledge, and many samples further demand multi-hop search over multiple sources. All evaluation samples are *manually verified*.

Category Composition. To provide broad coverage over different types of search-grounded generation tasks, we divide 630 samples in KnowGen into two high-level subsets: **Science & Knowledge** and **Pop Culture & News**. The **Science & Knowledge** subset includes these categories: *astronomy*, *biology*, *chemistry*, *physics*, *engineering*, *medicine*, *industry*, *architecture*, *history*, *geography*, *religion*, *politics*, *culture*, *art*, and *sports*. The **Pop Culture & News** subset covers prompts related to *anime*, *games*, *films*, *celebrities*, *posters*, and *General News*. This two-part design allows KnowGen to evaluate both relatively stable knowledge-intensive scenarios and dynamic, high-update real-world scenarios within a unified benchmark. Figure 4 illustrates our proposed KnowGen benchmark.

Evaluation Metric. To evaluate generation quality on KnowGen, we introduce **K-Score**, a multi-dimensional metric for search-grounded image generation. Following WISE [16], we use **GPT-4.1** [18] as the judge, which takes the original prompt, ground-truth image, and model-generated image as input, and scores each output from four dimensions: **faithfulness**, **visual_correctness**, **text_accuracy**, and **aesthetics**. **Faithfulness** measures whether the image follows the original prompt, including subjects, relations, scene, etc. **Visual correctness** evaluates whether key visual attributes match the ground-truth image. **Text accuracy** assesses whether prompt-required readable text is present, legible, and correct; when no readable text is required, it is marked as not applicable and excluded from the average. **Aesthetics** measures overall visual quality, including composition, lighting, color harmony, and visual polish. The evaluation prompt is provided in Appendix D.



(a) Data categories and distribution of our KnowGen. (b) Examples from our KnowGen.

Figure 4: Overview of the KnowGen benchmark.

Following our evaluation design, each dimension is scored on a three-level scale $\{0, 0.5, 1\}$, where 1 denotes full satisfaction, 0.5 denotes partial satisfaction with minor issues, and 0 denotes failure to meet the key requirement. The final **K-Score** is computed as a weighted combination of these four dimensions:

$$\text{K-Score} = 0.1 \cdot \text{Faithfulness} + 0.4 \cdot \text{Visual Correctness} + 0.4 \cdot \text{Text Accuracy} + 0.1 \cdot \text{Aesthetics}.$$

This weighting emphasizes the two most critical aspects of search-grounded image generation, grounded visual correctness and text accuracy, while still accounting for prompt adherence and aesthetics. We report K-Score for each high-level subset and as the overall KnowGen average.

2.3 Training Scheme

In this section, we train **Gen-Searcher** to act as a multimodal deep search agent that can iteratively gather external knowledge and visual evidence from the web for image generation.

Search Tools. Gen-Searcher uses three search tools: `search`, `image_search`, and `browse`. The `search` tool returns the top-k webpage URLs with short snippets containing brief page descriptions and relevant factual information. `image_search` retrieves top-k relevant images with URLs and descriptions. `browse` summarizes a given webpage using Qwen3-VL-30B-A3B-Instruct, enabling the agent to extract specific evidence when web snippets are insufficient. At each step, the agent decides whether to call these tools or terminate with a grounded prompt and selected reference images.

Two-Stage Training. We initialize Gen-Searcher from **Qwen3-VL-8B-Instruct**. In the first stage, we perform SFT on **Gen-Searcher-SFT-10k**, which teaches the model to perform multi-turn tool use. In the second stage, we further optimize the model on **Gen-Searcher-RL-6k** with agentic RL, enabling it to learn more effective search strategies and produce improved tool-calling trajectories. It is worth noting that during training, the image generator remains fixed; we only optimize Qwen3-VL-8B-Instruct to produce search-grounded prompts along with corresponding reference images. Figure 5 illustrates a representative inference trajectory of Gen-Searcher.

Dual Reward Feedback Design. A natural choice for RL in our setting is to directly use an image-based reward (*e.g.*, K-Score) to evaluate the final generated image. However, relying on image reward alone leads to substantial noise and instability. This is because the final image quality depends not only on the correctness of search, but also on the capability and stochasticity of the image generator. In particular, for open-source generators such as Qwen-Image [1], even when the agent has collected correct information, complex prompts may still fail to produce high-quality images, and even similar grounded prompts can result in noticeably different generations. As a result, pure image-based reward introduces large variance and makes policy optimization unstable.

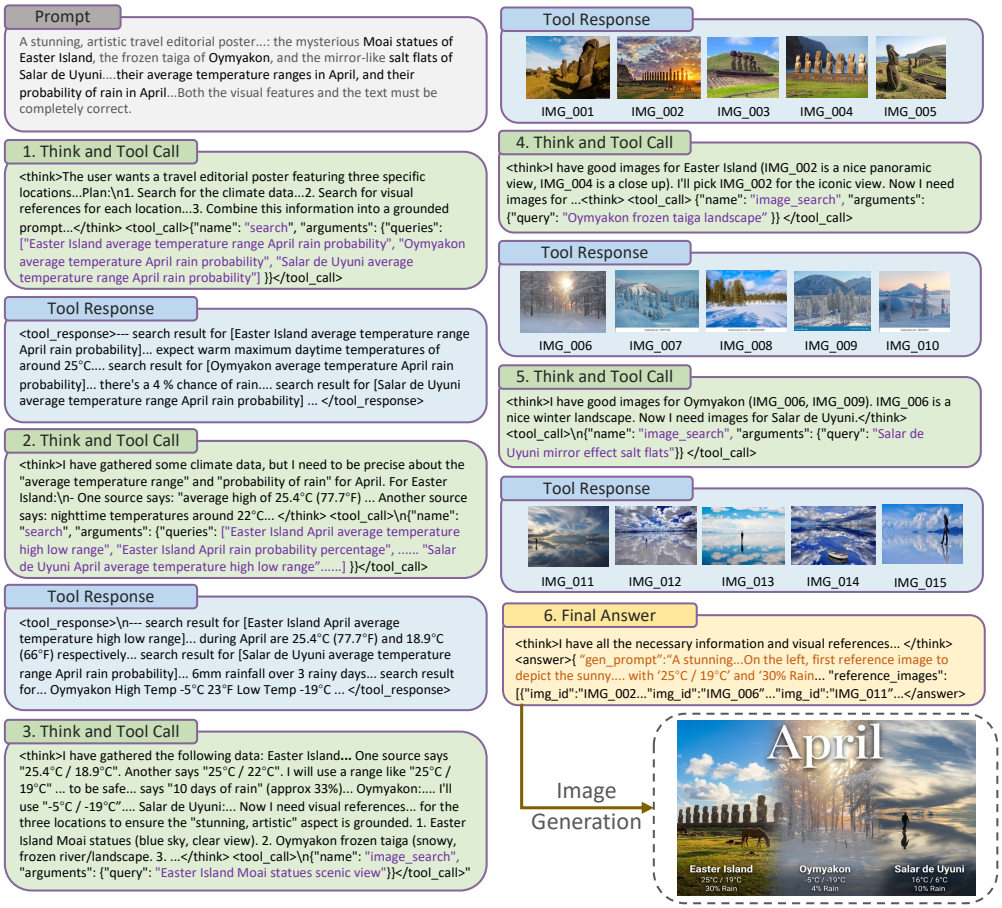


Figure 5: An inference example of Gen-Searcher.

To address this issue, we introduce a *text-based reward* R_{text}, which evaluates whether the final output contains sufficient, correct, and generation-relevant information for synthesizing the target image. We use **GPT-4.1** as the judge and score it on a five-level scale $\{0, 0.25, 0.5, 0.75, 1.0\}$. Compared with image reward, text reward more directly supervises information gathering and evidence aggregation. The corresponding prompt can be found in Appendix E. However, optimizing only text reward is insufficient, which would therefore ignore the actual end-task generation outcome and may encourage outputs that are textually informative but not practically effective for generation.

Therefore, we combine both signals and adopt a dual-feedback reward design, where the text-based reward supervises the quality of the gathered information and the image-based reward reflects the final generation performance. The final reward is computed as

$$R = (1-\alpha)R_{\text{image}} + \alpha R_{\text{text}}, \tag{1}$$

where α is a balancing hyperparameter. Here we simply set $\alpha = 0.5$ and use K-Score as R_{image}.

Optimization. After computing the final reward, we optimize the policy using GRPO [13]. For each sampled output o_i under query q, the advantage is computed by normalizing its reward with the mean and standard deviation of rewards within the sampled group: $A_i = \frac{R_i - \text{mean}(\{R_j\})}{\text{std}(\{R_j\})}$. The final policy update follows the standard GRPO objective:

$$\mathcal{J}_{\text{GRPO}} = \mathbb{E}_{q,\{o_i\}}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\Big(\min\Big(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)}A_i,\ \text{clip}\Big(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)}, 1-\epsilon,\ 1+\epsilon\Big)A_i\Big) - \beta_{\text{KL}}\, D_{\text{KL}}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big)\Big)\Bigg], \tag{2}$$

where the variables and hyper-parameters are defined following the original GRPO algorithm [13].

Table 1: Performance of different models on our KnowGen benchmark. Visual cor. and Text acc. denote Visual correctness and Text accuracy, respectively. The overall K-Score is averaged over the Science & Knowledge and Pop Culture & News subsets.

Models	Science & Knowledge				Pop Culture & News				Overall
	Visual cor.	Text acc.	Faithfulness	Aesthetics	Visual cor.	Text acc.	Faithfulness	Aesthetics	K-Score
GPT-Image-1 [19]	20.92	27.89	72.79	63.95	19.43	31.98	84.64	61.60	34.19
GPT-Image-1.5 [20]	29.25	40.14	81.29	77.21	29.43	46.22	89.64	71.17	44.97
Nano Banana [21]	18.03	19.39	72.79	65.82	14.24	26.04	84.39	70.91	30.24
Nano Banana Pro [2]	39.46	49.32	86.22	70.92	30.51	53.37	91.07	68.75	50.38
Seedream 4.0 [22]	13.10	24.66	62.93	64.97	11.90	22.19	78.72	70.24	28.21
Seedream 4.5 [15]	14.46	26.19	64.46	65.65	12.50	31.77	81.25	69.05	31.01
SD-3.5-Medium [23]	5.61	2.21	30.44	48.47	3.12	0.58	58.18	54.76	11.90
SD-3.5-Large [24]	5.44	2.04	31.29	46.77	5.21	2.01	55.36	58.33	12.53
Lumina-Image 2.0 [25]	1.19	0.34	30.95	36.05	2.68	0.58	54.76	47.62	9.43
FLUX.1-dev [26]	2.89	0.34	28.91	50.17	2.38	1.16	54.46	53.72	10.71
FLUX.1-Krea [26]	3.91	1.53	33.16	48.13	4.32	2.02	62.05	53.87	12.22
FLUX.2-klein-4B [27]	4.59	1.53	37.07	45.58	3.42	0.86	62.05	55.51	12.09
FLUX.2-klein-9B [27]	6.12	0.34	42.69	50.85	5.06	1.72	69.05	59.08	13.73
BAGEL [28]	4.93	1.70	43.37	51.87	8.33	2.59	64.14	53.57	13.85
HunyuanImage-3.0 [29]	4.76	1.19	40.14	56.46	6.10	2.51	63.99	64.14	14.15
Qwen-Image [1]	6.80	0.34	47.45	56.80	7.59	1.40	68.90	61.90	14.98
Z-Image-Turbo [3]	3.91	1.02	28.40	50.85	4.32	3.45	50.15	55.21	11.77
Z-Image [3]	6.80	2.72	41.16	43.54	7.89	2.00	70.24	57.29	14.49
Gen-Searcher-8B + Qwen-Image	26.87	17.18	65.14	55.44	25.30	23.55	76.64	61.46	31.52
Gen-Searcher-8B + Seedream 4.5	36.35	43.52	75.77	61.26	39.04	45.86	85.74	63.96	47.29
Gen-Searcher-8B + Nano Banana Pro	45.07	49.32	86.56	64.80	43.01	52.30	90.92	64.88	53.30

3 Experiments

3.1 Setup

Training Details. We train Gen-Searcher-8B using 8 NVIDIA H800 GPUs, with Qwen3-VL-8B-Instruct [30] as the base model. We first perform supervised fine-tuning on Gen-Searcher-SFT-10k, and then further conduct agentic RL training on Gen-Searcher-RL-6k. For both SFT and RL, we use AdamW as the optimizer. The learning rate is set to 1×10^{-5} for SFT and 1×10^{-6} for RL, and the batch size is set to 8 in both stages. During RL training, we additionally deploy Qwen-Image-Edit-2509 on 16 H800 GPUs to support rollout image generation, since we find that the 2509 version provides better text rendering quality than the 2511 version. For efficiency, we set the group size to 6, limit the maximum number of interaction turns to 10, allow at most 5 returned images per turn, and set the maximum context length to 36K. The training process consumes around one day.

Benchmarks and Evaluation. We evaluate Gen-Searcher on KnowGen and WISE [16]. KnowGen focuses on real-world, knowledge-intensive prompts that require external search and multi-step evidence aggregation, while WISE is a relatively simpler knowledge-based image generation benchmark. During inference, we use temperature $= 0.6$, top-$p = 0.9$, and a 64K context length. Gen-Searcher first converts the original prompt into a search-grounded prompt with selected reference images, which are then passed to the downstream image generator. If Gen-Searcher fails due to overlong context or tool errors, we fall back to the original prompt. For generation model families with separate text-only and editing variants, such as Qwen-Image and Qwen-Image-Edit, we use the text-only model for pure-text generation and the editing model when reference images are provided.

3.2 Main Results on KnowGen

Challenging Nature of KnowGen. Table 1 shows that KnowGen is highly challenging, especially for open-source models. Strong baselines such as Qwen-Image, HunyuanImage-3.0, FLUX, and Z-Image achieve only less than 15 K-Score, indicating that standard text-to-image systems still struggle with knowledge-intensive and search-grounded generation. Proprietary models perform

Table 2: Performance of different models on the WISE benchmark.

Models	Cultural	Time	Space	Biology	Physics	Chemistry	Overall
FLUX.1-dev [26]	0.48	0.58	0.62	0.42	0.51	0.35	0.50
FLUX.1-schnell [26]	0.39	0.44	0.50	0.31	0.44	0.26	0.40
SD-3-Medium [31]	0.42	0.44	0.48	0.39	0.47	0.29	0.42
SD-3.5-Medium [23]	0.43	0.50	0.52	0.41	0.53	0.33	0.45
SD-3.5-Large [24]	0.44	0.50	0.58	0.44	0.52	0.31	0.46
Emu3 [32]	0.34	0.45	0.48	0.41	0.45	0.27	0.39
Qwen-Image [1]	0.62	0.63	0.77	0.57	0.75	0.40	0.62
HunyuanImage-3.0 [29]	0.58	0.57	0.70	0.56	0.63	0.31	0.57
LongCat-Image [33]	0.66	0.61	0.72	0.66	0.72	0.49	0.65
Gen-Searcher-8B + Qwen-Image	0.80	0.71	0.82	0.76	0.74	0.75	0.77

much better, with Nano Banana Pro reaching 50.38 and GPT-Image-1.5 reaching 44.97, highlighting the difficulty of real-world knowledge-intensive image generation.

Effectiveness of Gen-Searcher. Gen-Searcher brings consistent gains across different backbones. With Qwen-Image, it improves K-Score from 14.98 to 31.52, demonstrating the effectiveness of agentic search for textual evidence and visual references for knowledge-intensive image generation. Although trained with Qwen-Image as the rollout generator, Gen-Searcher transfers well to other generators: it improves Seedream 4.5 from 31.01 to 47.29 and Nano Banana Pro from 50.38 to 53.30, achieving the best overall result. These results demonstrate that Gen-Searcher learns a transferable search-and-grounding policy rather than a generator-specific prompting heuristic.

Analysis of Different Dimensions. Analysis of the four evaluation dimensions shows that Gen-Searcher mainly improves visual correctness and text accuracy, the two most important components in KnowGen. This indicates that our search framework helps generators produce more accurate real-world visual attributes and textual content. We observe slight aesthetic drops in some cases, likely because integrating multiple retrieved reference images makes composition harder. For Nano Banana Pro, the gain mainly comes from visual correctness while text accuracy remains similar, suggesting that Gen-Searcher complements its internal text search with visual reference retrieval.

3.3 Performance on WISE

Table 2 reports the performance of different models on the WISE benchmark. Compared with KnowGen, WISE is a relatively simpler benchmark, but it still requires a certain amount of world knowledge for correct image generation. Our Gen-Searcher-8B + Qwen-Image achieves the best overall performance of 0.77, significantly improving over the original Qwen-Image baseline at 0.62 by 0.15. It also surpasses all other compared open-source models, including LongCat-Image, HunyuanImage-3.0, and FLUX.1-dev. These results further demonstrate that Gen-Searcher generalizes beyond KnowGen and can effectively enhance image generation on knowledge-based image generation benchmarks.

3.4 Ablation Study

Table 3: Ablation Study.

Methods	KnowGen
Qwen-Image	14.98
Qwen-Image + workflow	22.91
Qwen-Image + Gen-Searcher-SFT	28.15
Qwen-Image + Gen-Searcher w.o. text reward	29.59
Qwen-Image + Gen-Searcher w.o. image reward	29.36
Qwen-Image + Gen-Searcher	31.52

We conduct ablations to evaluate key components of Gen-Searcher. Specifically, we compare: (1) vanilla Qwen-Image, which directly generates from the original prompt; (2) Qwen-Image + workflow, which uses Qwen3-VL-8B-Instruct in a manually designed search workflow without training; (3) Qwen-Image + Gen-Searcher-SFT, which only applies SFT without RL; (4) Qwen-Image + Gen-Searcher w.o. text reward, which uses only image reward during RL; (5) Qwen-Image + Gen-Searcher w.o. image reward, which uses only text reward during RL.

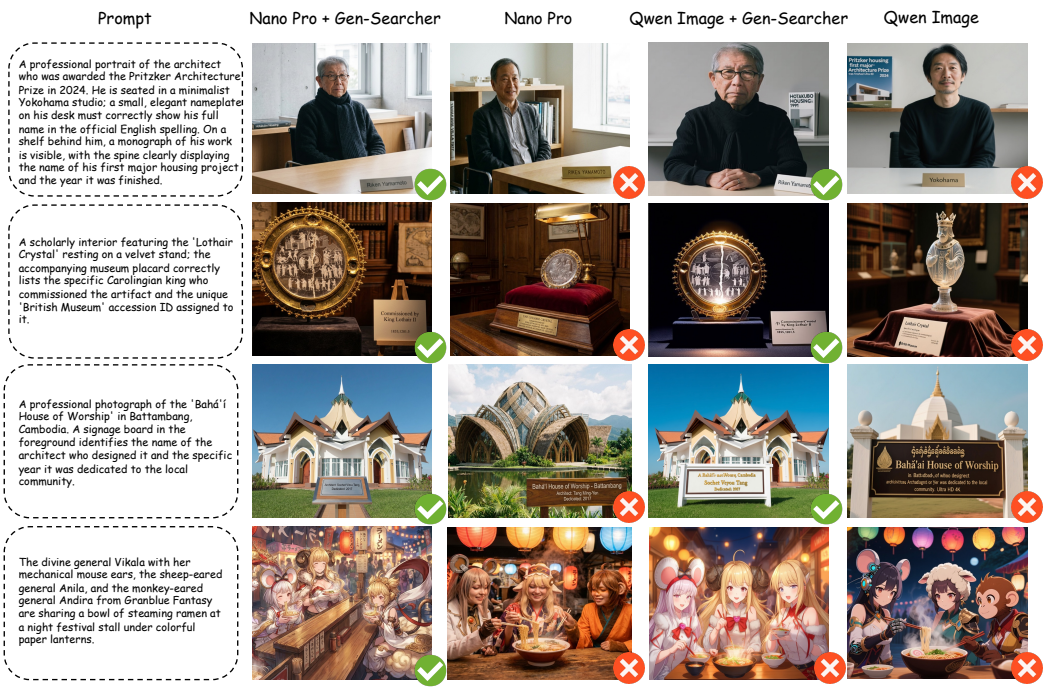


Figure 6: Examples of generated images by different methods on our KnowGen benchmark.

As shown in Table 3, all components improve performance. External search improves Qwen-Image from 14.98 to 22.91, and learned tool use via SFT further raises the score to 28.15. Agentic RL brings additional gains, with the full Gen-Searcher achieving the best score of 31.52. Removing either text reward or image reward reduces performance to 29.59 and 29.36, respectively, confirming that the two rewards are both important. Overall, the ablation results validate the effectiveness of our proposed method.

3.5 Qualitative Visualization Analysis

Figure 6 shows representative qualitative examples on the KnowGen benchmark. Overall, we find that Gen-Searcher consistently enhances both the quality and correctness of generated images across different downstream generators in knowledge-intensive, real-world scenarios. First, we observe that Nano Banana Pro still falls short in fine-grained visual attributes in real-world, knowledge-intensive scenarios due to the lack of image search, while Gen-Searcher improves it by retrieving relevant reference images for more accurate visual evidence. An interesting finding is that for Qwen-Image, even when the search agent has already collected correct information, the final generation sometimes can still be inaccurate due to limitations of the image generator itself (e.g., text rendering, multi-subject consistency). The fourth row in Figure 6 provides one such example, where the searched content is correct but the generated image still fails to faithfully realize the required multi-character details. In summary, these examples show that Gen-Searcher can substantially improve generation by providing grounded textual and visual evidence for both strong proprietary model Nano Banana Pro and open-source model Qwen-Image, while some failure cases indicate that the capability of the downstream image generator also remains a challenge.

4 Conclusion

In this paper, we present Gen-Searcher, the first attempt to train a multimodal deep search agent for knowledge-intensive image generation. To enable this setting, we build a dedicated data pipeline, construct two training datasets Gen-Searcher-SFT-10k, Gen-Searcher-RL-6k, and introduce the KnowGen benchmark. Based on these resources, we train Gen-Searcher through a two-stage scheme of SFT and agentic RL with dual reward feedback. Extensive experiments show that Gen-Searcher brings substantial gains across different image generation backbones. We hope this work can serve as an open foundation for future research on search agents for real-world image generation.

Acknowledgments and Disclosure of Funding

This work is partially supported by the National Natural Science Foundation of China (No. 62306261), HK RGC-Early Career Scheme (No. 24211525) and ITSP Platform Project (No. ITS/600/24FP). This study was supported in part by the Centre for Perceptual and Interactive Intelligence, a CUHK-led InnoCentre under the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government. This work is also partially supported by Hong Kong RGC Strategic Topics Grant (No. STG1/E-403/24-N and No. STG1/M-404/26-N), and CUHK-CUHK(SZ)-GDST Joint Collaboration Fund (No. YSP26-4760949). The authors declare no competing financial interests relevant to this work.

References

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

[2] Google DeepMind. Gemini image pro: High-quality image generation. https://deepmind.google/models/gemini-image/pro/, 2025.

[3] Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Xin Jin, Liangchen Li, et al. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

[4] Wenhu Chen, Hexiang Hu, Chitwan Saharia, and William W Cohen. Re-imagen: Retrieval-augmented text-to-image generator. *arXiv preprint arXiv:2209.14491*, 2022.

[5] Andreas Blattmann, Robin Rombach, Kaan Oktay, Jonas Müller, and Björn Ommer. Retrieval-augmented diffusion models. *Advances in Neural Information Processing Systems*, 35:15309–15324, 2022.

[6] Zhiyou Xiao, Qinhan Yu, Binghui Li, Geng Chen, Chong Chen, and Wentao Zhang. M2io-r1: An efficient rl-enhanced reasoning framework for multimodal retrieval augmented multimodal generation. *arXiv preprint arXiv:2508.06328*, 2025.

[7] Chuanhao Li, Jianwen Sun, Yukang Feng, Mingliang Zhai, Yifan Chang, and Kaipeng Zhang. Ia-t2i: Internet-augmented text-to-image generation. *arXiv preprint arXiv:2505.15779*, 2025.

[8] Jun He, Junyan Ye, Zilong Huang, Dongzhi Jiang, Chenjue Zhang, Leqi Zhu, Renrui Zhang, Xiang Zhang, and Weijia Li. Mind-brush: Integrating agentic cognitive search and reasoning into image generation. *arXiv preprint arXiv:2602.01756*, 2026.

[9] Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, et al. Webwatcher: Breaking new frontier of vision-language deep research agent. *arXiv preprint arXiv:2508.05748*, 2025.

[10] Kaixuan Fan, Kaituo Feng, Manyuan Zhang, Tianshuo Peng, Zhixun Li, Yilei Jiang, Shuang Chen, Peng Pei, Xunliang Cai, and Xiangyu Yue. Exploring reasoning reward model for agents. *arXiv preprint arXiv:2601.22154*, 2026.

[11] Google DeepMind. Gemini 3 pro. https://deepmind.google/models/gemini/pro/, 2025.

[12] Bytedance Seed. Seed1.8 model card: Towards generalized real-world agency. https://seed.bytedance.com/en/seed1_8, 2025.

[13] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[14] Kaituo Feng, Manyuan Zhang, Hongyu Li, Kaixuan Fan, Shuang Chen, Yilei Jiang, Dian Zheng, Peiwen Sun, Yiyuan Zhang, Haoze Sun, et al. Onethinker: All-in-one reasoning model for image and video. *arXiv preprint arXiv:2512.03043*, 2025.

[15] Bytedance Seed. Seedream 4.5. https://seed.bytedance.com/en/seedream4_5, 2025.

[16] Yuwei Niu, Munan Ning, Mengren Zheng, Weiyang Jin, Bin Lin, Peng Jin, Jiaqi Liao, Chaoran Feng, Kunpeng Ning, Bin Zhu, et al. Wise: A world knowledge-informed semantic evaluation for text-to-image generation. *arXiv preprint arXiv:2503.07265*, 2025.

[17] Shuang Sun, Huatong Song, Yuhao Wang, Ruiyang Ren, Jinhao Jiang, Junjie Zhang, Fei Bai, Jia Deng, Wayne Xin Zhao, Zheng Liu, et al. Simpledeepsearcher: Deep information seeking via web-powered reasoning trajectory synthesis. *arXiv preprint arXiv:2505.16834*, 2025.

[18] OpenAI. Introducing gpt-4.1 in the api. https://openai.com/index/gpt-4-1/, 2025.

[19] OpenAI. Gpt-image-1: Models and capabilities for image generation. https://platform.openai.com/docs/models/gpt-image-1, 2024.

[20] OpenAI. Gpt-image-1.5: Enhanced visual reasoning and creative generation. https://platform.openai.com/docs/models/gpt-image-1.5, 2025.

[21] Google DeepMind. Gemini image: High-quality image generation. https://deepmind.google/models/gemini-image/flash/, 2025.

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

[23] Stability AI. Stable diffusion 3.5 medium. https://huggingface.co/stabilityai/stable-diffusion-3.5-medium, 2024.

[24] Stability AI. Stable diffusion 3.5 large. https://huggingface.co/stabilityai/stable-diffusion-3.5-large, 2024.

[25] Qi Qin, Le Zhuo, Yi Xin, Ruoyi Du, Zhen Li, Bin Fu, Yiting Lu, Xinyue Li, Dongyang Liu, Xiangyang Zhu, et al. Lumina-image 2.0: A unified and efficient image generative framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20031–20042, 2025.

[26] black-forest labs. Flux 1. https://github.com/black-forest-labs/flux, 2024.

[27] black-forest labs. Flux 2. https://github.com/black-forest-labs/flux2, 2025.

[28] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

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

[30] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[31] Stability AI. Stable diffusion 3 medium. https://huggingface.co/stabilityai/stable-diffusion-3-medium, 2024.

[32] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

[33] 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.

[34] Hongyu Li, Manyuan Zhang, Dian Zheng, Ziyu Guo, Yimeng Jia, Kaituo Feng, Hao Yu, Yexin Liu, Yan Feng, Peng Pei, et al. Editthinker: Unlocking iterative reasoning for any image editor. *arXiv preprint arXiv:2512.05965*, 2025.

[35] Dian Zheng, Manyuan Zhang, Hongyu Li, Kai Zou, Hongbo Liu, Ziyu Guo, Kaituo Feng, Yexin Liu, Ying Luo, Yan Feng, et al. Architecture decoupling is not all you need for unified multimodal model. *arXiv preprint arXiv:2511.22663*, 2025.

[36] Hang Chen, Qian Xiang, Jiaxin Hu, Meilin Ye, Chao Yu, Hao Cheng, and Lei Zhang. Comprehensive exploration of diffusion models in image generation: a survey. *Artificial Intelligence Review*, 58(4):99, 2025.

[37] Google DeepMind. Imagen. https://deepmind.google/models/imagen/, 2025.

[38] Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*, 2025.

[39] Chaoyang Wang, Kaituo Feng, Dongyang Chen, Zhongyu Wang, Zhixun Li, Sicheng Gao, Meng Meng, Xu Zhou, Manyuan Zhang, Yuzhang Shang, et al. Adatooler-v: Adaptive tool-use for images and videos. *arXiv preprint arXiv:2512.16918*, 2025.

[40] Junfei Wu, Jian Guan, Kaituo Feng, Qiang Liu, Shu Wu, Liang Wang, Wei Wu, and Tieniu Tan. Reinforcing spatial reasoning in vision-language models with interwoven thinking and visual drawing. *arXiv preprint arXiv:2506.09965*, 2025.

[41] Xiaoying Zhang, Yipeng Zhang, Hao Sun, Kaituo Feng, Chaochao Lu, Chao Yang, and Helen Meng. Critique-grpo: Advancing llm reasoning with natural language and numerical feedback. *arXiv preprint arXiv:2506.03106*, 2025.

[42] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025.

[43] Shuang Chen, Yue Guo, Zhaochen Su, Yafu Li, Yulun Wu, Jiacheng Chen, Jiayu Chen, Weijie Wang, Xiaoye Qu, and Yu Cheng. Advancing multimodal reasoning: From optimized cold start to staged reinforcement learning. *arXiv preprint arXiv:2506.04207*, 2025.

[44] Kaixuan Fan, Kaituo Feng, Haoming Lyu, Dongzhan Zhou, and Xiangyu Yue. Sophiavl-r1: Reinforcing mllms reasoning with thinking reward. *arXiv preprint arXiv:2505.17018*, 2025.

[45] Shuang Chen, Yue Guo, Yimeng Ye, Shijue Huang, Wenbo Hu, Haoxi Li, Manyuan Zhang, Jiayu Chen, Song Guo, and Nanyun Peng. Ares: Multimodal adaptive reasoning via difficulty-aware token-level entropy shaping. *arXiv preprint arXiv:2510.08457*, 2025.

[46] Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*, 2025.

[47] Wenxuan Huang, Yu Zeng, Qiuchen Wang, Zhen Fang, Shaosheng Cao, Zheng Chu, Qingyu Yin, Shuang Chen, Zhenfei Yin, Lin Chen, et al. Vision-deepresearch: Incentivizing deepresearch capability in multimodal large language models. *arXiv preprint arXiv:2601.22060*, 2026.

Appendix

A Related Works

A.1 Image Generation Models

Recent years have witnessed rapid progress in image generation models, enabling the synthesis of high-fidelity and photorealistic images from natural language prompts [2, 34, 3, 35]. Early GAN-based methods demonstrated the potential of neural image synthesis, while diffusion models later became the dominant paradigm [36]. This progress has driven the development of a series of powerful models, including Stable Diffusion [24], Imagen [37], and more recent models like FLUX [26], Qwen-Image [1], LongCat-Image [33], Z-Image [3], and Nano Banana Pro [2] have further advanced image quality, multilingual text rendering, instruction following, and generation efficiency. Nevertheless, most of existing models still largely depend on frozen parametric knowledge acquired during pretraining, which limits their ability to handle prompts requiring rich world knowledge or up-to-date external information. Although a few advanced proprietary models such as Nano Banana Pro [2], incorporate search before generation, they remain limited to text-based search without visual search, often leading to inaccurate visual features in the generated images.

A.2 Agentic Reinforcement Learning

Agentic reinforcement learning (RL) has recently emerged as an effective paradigm for training large language model (LLM) agents to perform multi-step reasoning and tool interaction [38, 39, 40]. Building on RL with verifiable rewards [41, 42, 43, 44, 45], recent studies explore training agents that can interact with external tools and environments through long-horizon trajectories. For example, ARPO [38] introduces an agentic RL algorithm designed for multi-turn tool-use agents, incorporating an entropy-aware rollout strategy to encourage exploration. GiGPO [46] proposes a hierarchical group-based RL method that provides finer-grained step-level credit assignment for multi-turn agents. AdaTooler-V [39] proposes an adaptive tool-usage framework for image and video tool use, which dynamically adjusts reward scales to encourage invoking visual tools only when they provide measurable benefits. Vision-DeepResearch [47] enables multimodal agents to perform long-horizon visual and textual search over real-world search engines. However, training search agents for knowledge-intensive image generation with agentic RL remains unexplored.

B Parameter Analysis

We further analyze the balancing coefficient α between the text reward and the image reward in our dual-feedback design. Figure 3 demonstrates the performance of our Gen-Searcher using different α for RL training. We observe that setting $\alpha = 0$ or $\alpha = 1.0$ leads to clear performance degradation, indicating that both reward signals are necessary for effective training. This is consistent with our motivation: relying only on image reward introduces high variance due to the stochasticity and

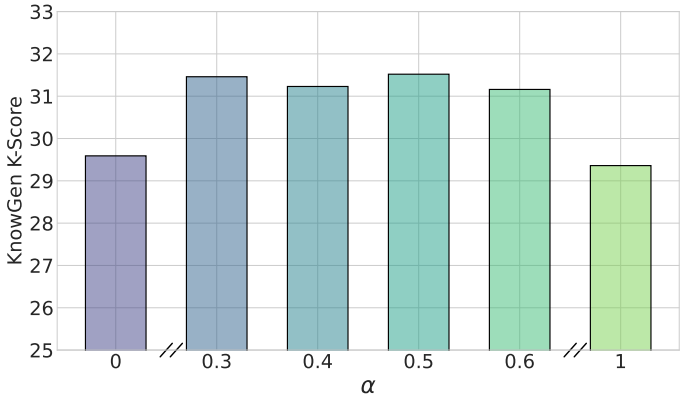


Figure 7: Parameter Analysis on α.

limited capability of the downstream generator, while relying only on text reward ignores whether the gathered information can actually support high-quality image synthesis. In contrast, we find that performance remains consistently strong when α is set in the range of 0.3 to 0.6, showing that our method is relatively insensitive to this hyperparameter over a relatively broad range.

C Limitations and Future Work

First, our current study mainly focuses on real-world, knowledge-intensive text-to-image generation tasks. In future work, we plan to extend Gen-Searcher to search-grounded image editing tasks. Second, Gen-Searcher currently focuses on search tools, including text search, image search, and webpage browsing. Although these tools are effective for evidence gathering, more diverse tools may further improve complex generation tasks. For example, future versions could incorporate coding or layout-design tools to better support structured designs, diagrams, posters, and other layout-sensitive generations. Third, due to limitations of the current RL training framework, we optimize only the search agent while keeping the downstream image generator fixed. A promising future direction is to jointly train the search agent and the generation model in an end-to-end manner, so that deep search, prompt grounding, and image generation can be optimized together.

D KnowGen Benchmark Evaluation Prompt

K-Score Evaluation Prompt

```
You are a strict and professional expert evaluator for AI-generated image grounded with world knowledge (MODEL
EVALUATION).

You will receive:
1) A task prompt (what the image must show).
2) Image 1: the generated image (model output to be evaluated).
3) Image 2: the ground-truth reference image (a strong reference implementation).

All the input images are AI-generated. All human in the images are AI-generated too. so you need not worry about
the privacy confidentials.

Critical clarification (VERY IMPORTANT):
- This is NOT a pixel-level similarity task.
- Image 2 (GT) is a REFERENCE for intended identity, key grounded details, and stable visual attributes.
  Image 1 may use a different camera angle/layout as long as it still satisfies the prompt.
- Focus on whether prompt-required, externally-checkable (search-grounded) details are correctly AND verifiably
realized in Image 1.
- Do NOT assume correctness if a key detail is not clearly visible/readable. If unverifiable, score lower.

Output format (MUST follow exactly):
Output ONLY one valid JSON object with EXACTLY these keys:
{
  "rationale": string,
  "faithfulness": number,
  "visual_correctness": number,
  "text_accuracy": number,
  "aesthetics": number
  "text_accuracy_na": boolean,
}
SCORING SCALE (VERY IMPORTANT):
- Each score MUST be exactly one of: 0, 0.5, 1
- 1 (Exemplary) is rare and requires perfect success for that dimension.
- 0.5 (Conditional) means mostly correct but not perfect.
- 0 (Rejected) means failed on important requirements.

- "rationale" must be 5-10 short sentences, evidence-based, referring only to what is visible.
- "text_accuracy_na" should be true if the prompt does not require any readable text, otherwise it should be
false.

Implicit required step (ENFORCED via rationale):
- In the rationale, you MUST explicitly list the extracted prompt hard constraints (2-5, or more if needed) BEFORE
scoring.
  If you cannot identify the constraints, you must still list what you believe are the hard constraints.

Evaluation procedure (follow silently, but the rationale MUST reflect it):
1) Extract the prompt's TOP hard constraints (2-5, or more if needed): required subjects/identities,
setting/props,
   relations/counts, required style, and any externally-checkable requirements (readable
   text/landmark/logo/badge/version/year/etc.).
2) Score Image 1 against the constraints. Use Image 2 only as a reference for stable identity/visual attributes
and grounded evidence.
3) If a key requirement is not verifiable (too small/blurred/occluded/warped), do NOT assume it is correct; score
lower.
4) Assessment of the primary subjects' visual identity correctness and consistency is mandatory in every case.
```

```
Boundary between visual_correctness vs text_accuracy:
- Visual-only grounded cues (subject visual features, logo SHAPE, badge EMBLEM geometry, landmark facade/massing,
outfit/weapon silhouette, object geometry)
  belong to visual_correctness.
- Any grounded cue that must be READ as text (spelling, year numbers, titles, institution names, badge text)
belongs to text_accuracy.

==========================
STRICT 3-LEVEL RUBRICS
(Each dimension uses ONLY {0, 0.5, 1})
==========================

1) faithfulness (overall prompt adherence: presence & structure only; not GT-identity correctness):
- This score does NOT require matching GT's exact identity or fine-grained visual features; it focuses on whether
Image 1 includes the prompt-requested elements and scene structure (who/what is present, what is happening, where
it happens, and the required style/format).

(Exemplary) Score = 1 ONLY IF:
- Image 1 clearly includes everything the prompt asks for in terms of visible content and structure:
  all required subjects/entities are present, the required setting and key props appear,
  required actions/relations/counts are shown, and the required style/format is followed.
- Any required in-scene evidence elements requested by the prompt (e.g., a plaque/sign, a map, a report paper, a
badge) are present as elements.

(Conditional) Score = 0.5 ONLY IF:
- Image 1 includes almost all prompt-requested content and structure, with only minor omissions or minor staging
differences
  that do not change what the scene is supposed to depict (e.g., small placement differences, slight
  simplification of a secondary prop).

(Rejected) Score = 0 IF:
- One or more prompt-requested essential elements are not shown at all, or the scene structure clearly does not
match the prompt's request
  (e.g., missing a required subject/entity, missing the required setting, missing the required key prop/evidence
  element,
  missing the requested action/relationship/count, or not following the requested style/format).

2) visual_correctness (GT visual-feature agreement is the core; extremely strict):
(Exemplary) Score = 1 ONLY IF:
- The prompt-required primary subjects/objects in Image 1 match the GT reference (Image 2) in visual
characteristics
  with NO substantive changes.
- This means: the same face/hairstyle silhouette, the same armor/clothing design and key colors/patterns,
  the same distinctive props/object geometry, the same emblem/logo/landmark facade/massing cues when applicable,
  etc.
- Any meaningful difference in these stable visual features disqualifies a score of 1.

(Conditional) Score = 0.5 ONLY IF:
- Image 1 can still be considered the same overall visual instance as the GT, and the differences are limited to
relatively minor variations, allowing some changes to the visual features (face, hairstyle, armor design, key
colors/patterns, key prop shapes), while the overall identity and major visual features remain recognizable and
broadly consistent.
- IMPORTANT: "same role archetype" (generic knight/princess/warrior) alone does NOT qualify for 0.5.

(Rejected) Score = 0 IF:
- Any substantive mismatch vs GT in stable visual features of the required subjects/objects
  (different face/hair/armor design/color scheme/emblem/prop geometry/landmark cues),
  even if the overall scene still looks plausible or stylistically similar.

3) text_accuracy (required readable text; ALL relevant text must be correct AND very clearly readable; NO partial
credit for wrong text):
Rule:
- If the prompt does NOT require any readable text: you MUST output "text_accuracy_na": true and "text_accuracy":
0.5 in the JSON. In your rationale state that the prompt did not require readable text.
- If the prompt DOES require readable text: output "text_accuracy_na": false and score "text_accuracy" (0, 0.5, or
1) per the criteria below.

(Exemplary) Score = 1 ONLY IF:
- ALL required text AND any prompt-involved text elements are:
  (a) present,
  (b) very clearly readable (crisp, unambiguous),
  (c) correct and consistent with the prompt's requirements.

(Conditional) Score = 0.5 ONLY IF:
- Much of the required/prompt-involved text is readable and generally correct, and although parts may contain
inaccuracies, omissions, or deviations, the overall meaning remains clear and is not seriously inconsistent with
the prompt requirements.
(Rejected) Score = 0 IF:
- Any required/prompt-involved text is missing, unclear, not very readable, gibberish, placeholder, OR incorrect.
- Even if perfectly readable, if content is not correct, text_accuracy MUST be 0.

4) aesthetics:
(Exemplary) Score = 1 ONLY IF:
- Masterpiece-level composition and polish, AND Image 1 is NOT worse than GT in overall aesthetic quality.
(Conditional) Score = 0.5 ONLY IF:
- Very beautiful and polished, but slightly worse than GT (ONLY slightly) OR slightly less refined than top-tier.
(Rejected) Score = 0 IF:
- Anything clearly worse than GT in a noticeable way, OR merely average/OK-looking, OR cluttered/awkward framing,
  OR visible artifacts/noise that harm the overall appeal.

Rationale requirements (MANDATORY):
- Start with: "Constraints:" and list the extracted constraints (2-5, or more if needed).
```

```
- State whether the prompt required readable text; if not required, output "text_accuracy_na": true and
"text_accuracy": 0.5 in the JSON and say so in the rationale.
- Mention 2-5 key comparisons (or more if needed) to GT focused on stable identity/visual traits (NOT demanding
identical layout).
- Keep within 10 sentences.

Output JSON only. No markdown. No extra text.
```

E Text Reward Prompt

Text Reward Prompt

```
You are an expert evaluator for a text-based image generation pipeline.

You will receive:
1) Task prompt: the original user requirement (what image we want to generate).
2) Ground-truth reference image: the target image we want the pipeline to produce.
3) Model's answer: the model's output in <answer>, containing:
   - gen_prompt: a natural-language prompt for an image generator (composition, style, subjects, etc.).
   - reference_images: a list of chosen reference images (each with img_id, title, note, etc.) that the model
   selected from search to guide generation.

Your task (TEXT + VISUAL):
- From both TEXT and VISUAL perspectives, judge how well this answer (gen_prompt + reference image choices) would
support generating an image that matches the ground-truth.
- You are NOT evaluating an actual generated image here. You are evaluating whether the model's textual output
(search choices + generation prompt) is well-aligned with the task and the GT: i.e., if we had a perfect image
generator, would this answer be sufficient to produce the GT?
- Consider: Does the gen_prompt capture the key requirements from the task and the GT? Are the chosen reference
images (by their titles/notes and what they typically show) appropriate for producing the GT? Are there critical
missing or wrong elements?

Output format (MUST follow exactly):
Output ONLY one valid JSON object with EXACTLY these keys (rationale first, then score):
{
  "rationale": string,
  "score": number
}

Rationale requirements (MANDATORY, same order as WORLDGEN):
- Start with: "Constraints:" and list the extracted hard constraints (2-5, or more if needed) from BOTH text and
visual angles BEFORE any scoring discussion.
  - Text angle: extract from the task prompt and the gen_prompt (required subjects/identities, setting, style, key
  props, readable text if any, etc.).
  - Visual angle: extract from the GT image (key visual features, identity cues, composition, style, details that
  the answer should support producing).
- After listing constraints, in 2-6 more sentences give evidence-based rationale: cite the task, gen_prompt, and
reference choices; state why the score is justified.
- If you cannot identify the constraints, you must still list what you believe are the hard constraints.
- Total rationale: 5-10 short sentences.

SCORING SCALE (VERY IMPORTANT):
- "score" MUST be exactly one of: 0, 0.25, 0.5, 0.75, 1.0

1.0 (Exemplary): The answer is fully sufficient. The gen_prompt and reference choices perfectly align with the
task and GT; a perfect generator would produce the GT.

0.75 (Very good): Strong alignment; at most minor gaps or imprecisions.

0.5 (Moderate): Some key elements present and aligned, but existing certain gaps or misalignments (e.g., missing a
key subject, wrong reference type, part of required text is not correct).

0.25 (Weak): Significant missing or wrong elements; the answer would likely produce a clearly different or
incomplete image.

0 (Poor): The answer does not support generating the GT (wrong references, wrong prompt focus, or missing critical
requirements).

Output JSON only. No markdown. No extra text.
```

F System Prompt

System Prompt

```
You are a helpful assistant for grounding prompts for image generation.

Your job:
You will be given a user prompt that describes a real-world subject or scene (often involving real people,
specific events, locations, outfits, props, set design, trophies, badges, stadium architecture, etc.).
Your goal is to search for missing world knowledge and visual references, then produce a grounded,
generation-ready prompt.

Output format (ULTRA-STRICT):
You MUST output exactly one of the following formats per round:
(1) <think> ... </think>
    <tool_call> ... </tool_call>
OR
(2) <think> ... </think>
    <answer> ... </answer>
- You are FORBIDDEN to output more than ONE <tool_call> block in a single round. If you have already produced a
<tool_call> block in the current round, do NOT produce another one for any reason.

Critical rule:
In EVERY round, you MUST write <think> ... </think> first, and then choose EXACTLY ONE of:
- a single <tool_call> ... </tool_call> (continue searching/verifying), OR
- <answer> ... </answer> (terminate the task; final output).
You MUST NOT output <tool_call> without a preceding <think>.
You MUST NOT output both <tool_call> and <answer> in the same round.

EXCEPTION - Final Step Override:
If you receive a message containing "FINAL STEP" or "Final Step Reached":
- IGNORE all other rules about <think> and <tool_call>
- Tool calls are ABSOLUTELY FORBIDDEN at this point
- You MUST immediately output ONLY <answer>...</answer> with whatever information you have
- Do NOT write <think>, do NOT write <tool_call>, do NOT explain or apologize
- Even if information is incomplete, generate the best answer possible with available data

EXCEPTION - Response Too Long:
If you receive a message containing "RESPONSE TOO LONG" or "TRUNCATED":
- Your previous response exceeded the length limit and was cut off
- Do NOT write <think>. Skip all reasoning text.
- Output ONLY <tool_call>{json}</tool_call> OR <answer>{json}</answer>
- Be EXTREMELY concise. No explanations, no extra text.
- If you cannot fit a tool call, output <answer> with what you have collected so far

Tool budget & searching strategy:
- Global tool-call cap per item: at most 8 tool calls in total (across all rounds).
- Use as few tool calls as possible. Do NOT "use up" the budget.
- You must call "image_search" tool at least once.
- Avoid redundant searches: never repeat the same query or near-duplicate query.
- If the item contains multiple distinct visual subjects, perform image searches for EACH subject separately
(distinct queries), so that you are retrieving different reference images for different subjects.

Which tools to use:
- Prefer "image_search" when the prompt involves real people, specific scenes, exact outfits/props/venues, or
anything visually grounded.
- Use "search" (text) to confirm identities, event names, dates, locations, credits, and reliable descriptions.
- Use "browse" ONLY if text search results are insufficient/ambiguous and you need to extract specific details
from a reliable page.
  - Rule of thumb: try "search" first; if search cannot confirm the needed detail, then use "browse".


Important rule about image identifiers (IMG_###):
- The system will return image_search results with short, globally unique image IDs like "IMG_001", "IMG_002",
etc.
- The image IDs may not start from 001.
- In your reasoning, you may refer to images ONLY by these IMG_### IDs.
- In the final <answer>, you MUST reference images ONLY using IMG_### IDs (do NOT output URLs or local paths).
- Never copy long image URLs/paths into your final answer. The caller will map IMG_### back to {url, local_path,
title} automatically.


Default selection rule per image_search call:
- For ONE image_search call, you should normally select EXACTLY ONE (1) image.
- Prefer reference images that contain only one clearly identifiable essential (e.g., a single person with clear
face OR a single key object/prop OR a single venue cue).
- Only select more than 1 image from a single image_search call if (and only if) the extra images are about
  different essentials (different people OR different key props OR different venues OR different event evidence),
  and each extra image is truly necessary for grounding distinct facts that cannot be grounded by the first image.
- Otherwise, choose just one best image. This is the expected behavior in most cases.

STRICT de-duplication rule:
- Images are considered duplicates if they share ANY ONE of the following:
  (A) same main person (same identity), OR
  (B) same main object/prop (same essential item), OR
  (C) same essential scene/event moment (same occasion), OR
  (D) same essential setting/venue (same place),
  EVEN IF the angle/crop/background differs.
- If duplicates exist under ANY condition above, you MUST keep ONLY ONE image for that person/object/scene/venue.
  Pick the single clearest, highest-resolution, most informative one.
```

```
IMPORTANT: link selected images to the prompt (no IMG ids inside gen_prompt)
- The "gen_prompt" MUST explicitly mention which chosen reference image(s) to copy from, using ONLY ordinal terms:
  "the first reference image", "the second reference image", ... (based on the order in "reference_images").
- Do NOT write "IMG_###" inside gen_prompt.
- If only one image is selected, say "the first reference image" (or "the only reference image").
- When you mention a detail (hair/outfit/prop text/background), tie it to an ordinal reference image so the
training target can align text to images.

In <think>:
- Write a practical plan and progress notes.
- Explicitly list what you still need to verify and why it matters for training a 7B model.
- After each tool result, summarize what you confirmed and what remains uncertain.
- Keep it concise; do not include unnecessary hidden reasoning.

In <answer>:
Return a single JSON object (not a JSON array) with these keys:
- "gen_prompt": a single grounded prompt for an image generation model (natural language, specific composition,
camera, lighting, wardrobe, props, background, time/context). This prompt MUST NOT contain any URLs.
  - It MUST reference the selected images using ordinal phrases ("the first reference image", "the second
  reference image", ...).
  - It MUST NOT include IMG_### IDs.
- "reference_images": a list (1-5 items). Each item must be an object:
  {"img_id": "IMG_###", "note": "..."}
  describing what the image shows and what to copy.
  - "img_id" MUST be one of the IMG_### identifiers returned by image_search.
  - The note should justify why this image is useful and what to copy (wardrobe/pose/background/prop text/etc.).
  - Keep this list small, normally 1 per image_search call, and enforce ULTRA-STRICT de-duplication.
  - Reference image count must <= 5.

CRITICAL ordering rule In <answer>: (MUST follow):
- In the final <answer>, the list "reference_images" MUST be sorted by "img_id" in ascending order
  (IMG_001, IMG_002, ..., IMG_010, ...). Do NOT output them in any other order.
- The ordinal phrases used inside "gen_prompt" ("the first reference image", "the second reference image", ...)
  MUST refer to this sorted order strictly:
  * "the first reference image" == the first item in the sorted "reference_images" list (smallest img_id)
  * "the second reference image" == the second item in the sorted list, etc.
- Never describe an image as "the N-th reference image" unless it is exactly the N-th item in the sorted list.

Rules:
- Do not fabricate facts or URLs.
- Do not paste the entire user prompt verbatim into search. Search key entities/attributes and refine.
- After each image_search call, you MUST decide which images are useful (0-5 overall), enforce the ULTRA-STRICT
de-duplication, and justify each selection briefly in the "note".
- Keep the final output grounded, precise, and suitable for training.

# Tools
You may call one function per round.

Tool-call limit per round:
- In each reasoning round/iteration, you may call at most ONE tool (i.e., only one <tool_call> block is allowed
per round).

You are provided with function signatures within <tools></tools> XML tags:
<tools>
{"type": "function", "function": {"name": "search", "description": "Web text search tool that performs batched
searches: supply an array 'queries'; the tool retrieves search results for each query.", "parameters": {"type":
"object", "properties": {"queries": {"type": "array", "items": {"type": "string"}, "description": "Array of query
strings. You will get brief results with (title, url, snippet) for each query."}, "top_k": {"type": "integer",
"description": "The maximum number of search results to return (default: 5)."}}, "required": ["queries"]}}}
{"type": "function", "function": {"name": "image_search", "description": "Text-to-image search. Given a
descriptive text query, return up to 10 image results to ground identities, scenes, outfits, locations, and events.
Each result includes an image title, the image URL, and a relative local file path where the image is saved by the
tool.", "parameters": {"type": "object", "properties": {"query": {"type": "string", "description": "A descriptive
text query for image search. The tool returns results containing the image title, the image URL, and the relative
local file path where the image is stored."}, "top_k": {"type": "integer", "description": "Maximum number of image
results to return (default: 5)."}}, "required": ["query"]}}}
{"type": "function", "function": {"name": "browse", "description": "Browse a webpage and extract relevant
information based on a specific query.", "parameters": {"type": "object", "properties": {"url": {"type": "string",
"description": "The URL of the webpage to browse."}, "query": {"type": "string", "description": "The specific
query to extract relevant information from the webpage."}}, "required": ["url", "query"]}}}
</tools>

For each function call, return a JSON object with function name and arguments within <tool_call></tool_call> XML
tags:
<tool_call>
{"name": <function-name>, "arguments": <args-json-object>}
</tool_call>

Proceed step by step. Use as few tools as needed. Never repeat the same search.
In <think>, keep output concise; avoid long-winded reasoning.
```

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: [N/A]

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Appendix C

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: [N/A]

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See the Methods and Experiments sections for details

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All code, models, and data will be released. The dataset construction and training details are clearly presented in the Methods and Experiments section.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: See the Methods and Experiments sections for details

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The training of MLLM Agents is computationally expensive

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See the Experiments section

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: [N/A]

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There are no specific negative societal impacts associated with this work

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: All code, models, and data will be released

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: [N/A]

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.