GEMS: AGENT-NATIVE MULTIMODAL GENERATION WITH MEMORY AND SKILLS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent multimodal generation models have achieved remarkable progress on general-purpose generation tasks, yet continue to struggle with complex instructions and specialized downstream tasks. Inspired by the success of advanced agent frameworks such as Claude Code, we propose **GEMS** (Agent-Native Multimodal **GE**neration with **M**emory and **S**kills), a framework that pushes beyond the inherent limitations of foundational models on both general and downstream tasks. GEMS is built upon three core components. Agent Loop introduces a structured multi-agent framework that iteratively improves generation quality through closed-loop optimization. Agent Memory provides a persistent, trajectory-level memory that hierarchically stores both factual states and compressed experiential summaries, enabling a global view of the optimization process while reducing redundancy. Agent Skill organizes reusable domain guidance in a shared repository, allowing diverse tasks to benefit from relevant expertise within the same generation workflow. Across five mainstream tasks and four downstream tasks, evaluated on multiple generative backends, GEMS consistently achieves significant performance gains. Most notably, it enables the lightweight 6B model Z-Image-Turbo to match the performance of the state-of-the-art Nano Banana 2 on GenEval2, demonstrating the effectiveness of agent harness in extending model capabilities beyond their original limits. Code: `https://anonymous.4open.science/r/review-code-5d4224fd95dd/`

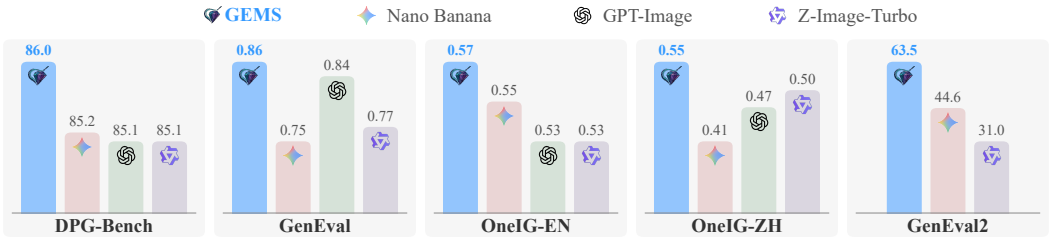


Figure 1: Overall Performance of GEMS with Z-Image-Turbo. GEMS enables a lightweight, distilled 6B model Z-Image-Turbo to outperform prominent closed-source models, such as Nano Banana and GPT-Image, across various mainstream benchmarks.

1 INTRODUCTION

Multimodal generation has undergone transformative growth in recent years Ramesh et al. (2022); Saharia et al. (2022); Brooks et al. (2024), where advanced algorithms Ho et al. (2020); Song et al. (2020); Ho & Salimans (2022); Lipman et al. (2022); Liu et al. (2022); Albergo & Vanden-Eijnden (2022); Chen et al. (2025) and architectural designs Rombach et al. (2022); Peebles & Xie (2023); Esser et al. (2024) have significantly enhanced the quality and accessibility of visual synthesis. Leading closed-source models, such as GPT-Image and Nano Banana, alongside prominent open-source frameworks like Qwen-Image Wu et al. (2025) and Z-Image Cai et al. (2025), have set new state-of-the-art records across various benchmarks. These models exhibit remarkable proficiency in handling mainstream and straightforward tasks Ghosh et al. (2023); Hu et al. (2024); Chang et al.

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

(2025), consistently producing high-fidelity results that align closely with general-purpose textual prompts. Despite these achievements, they often struggle when handling intricate, multi-faceted instructions Kamath et al. (2025) or specialized downstream applications Venkatesh et al. (2025); Wang et al. (2026); Geng et al. (2025), which constitutes the persistent "long-tail" challenge where general-purpose capabilities reach their limits.

To bridge these gaps, inference-time scaling Hao et al. (2023); Ma et al. (2025) has emerged as an important strategy for enhancing model performance. Current research explores iterative refinement loops Jiang et al. (2026b); Li et al. (2025a); Guo et al. (2025) and multi-agent collaborative systems Li et al. (2025b); Wan et al. (2025); Kovalev et al. (2025) to tackle complex tasks. Specialized multi-agent frameworks target downstream domains, such as creative drawing Venkatesh et al. (2025) and academic illustration Zhu et al. (2026), through domain-specific optimizations. However, existing approaches face two key limitations. First, they often lack an effective mechanism for organizing and reusing information across refinement steps. Frameworks such as Maestro Wan et al. (2025) primarily rely on successive updates, while other iterative approaches Jiang et al. (2026b); Li et al. (2025a) accumulate historical context without distilling useful guidance. These approaches risk either overlooking useful information from earlier steps or introducing redundancy. Second, domain-specific knowledge is often embedded in specialized workflows rather than organized for reuse. Although systems targeting particular downstream tasks Wang et al. (2026); Venkatesh et al. (2025); Zhu et al. (2026) achieve strong results within their domains, their specialized coordination mechanisms make it difficult to incorporate this knowledge into a generation pipeline or extend it to other tasks.

We propose **GEMS** (Agent-Native Multimodal **GE**neration with **M**emory and **S**kills), a framework that organizes generation trajectories and reusable domain knowledge within a unified optimization process. Inspired by memory and skills in agent frameworks such as Claude Code and OpenClaw, GEMS adapts these mechanisms to multimodal generation through three components: (1) **Agent Loop**, a multi-agent framework that decomposes complex instructions into evaluation criteria and iteratively refines generation through feedback; (2) **Agent Memory**, a persistent mechanism that treats factual artifacts and reasoning traces differently: it preserves generated images, prompts, and feedback while compressing reasoning into concise experiences. Unlike simple context accumulation Jiang et al. (2026b) or successive updates Wan et al. (2025), this design retains evidence of satisfied constraints alongside guidance for subsequent refinement, reducing redundancy while preserving optimization context; (3) **Agent Skill**, an extensible repository that separates domain knowledge from task-specific workflows Venkatesh et al. (2025); Zhu et al. (2026). By selecting relevant guidance for each prompt, it makes domain expertise available within the same refinement process across tasks. Together, these components connect domain guidance with accumulated experience, grounding refinement in task requirements and prior successes and failures.

To validate the effectiveness of GEMS, we conducted extensive experiments across nine distinct tasks, including five challenging mainstream benchmarks such as GenEval2 Kamath et al. (2025) and four specialized downstream tasks spanning diverse domains. We evaluated GEMS across multiple generative backends. Specifically, leveraging the lightweight, distilled Z-Image-Turbo Cai et al. (2025), GEMS yielded average score gains of 14.22 on mainstream benchmarks and 14.03 on downstream tasks. Most notably, our framework enables the 6B Z-Image-Turbo to match the performance of Nano Banana 2 on GenEval2, demonstrating that agentic reasoning and domain-specific expertise can improve the performance of the underlying generator. We further evaluated our framework on another mainstream open-source model, Qwen-Image-2512 Wu et al. (2025), where it also achieved substantial performance gains. These results demonstrate the effectiveness of our framework across different model architectures and scales.

In summary, our primary contributions are as follows:

- We propose **GEMS**, an agent-native multimodal generation framework that employs iterative refinement to significantly enhance performance in complex generation tasks.
- We introduce a persistent Agent Memory mechanism utilizing hierarchical compression, which efficiently manages historical context in multi-turn optimization trajectories.
- We introduce reusable domain skills into a shared generation workflow, enabling automatic access to relevant expertise across mainstream and downstream tasks.
- Extensive experiments across nine diverse tasks validate the effectiveness of GEMS, highlighting the transformative potential of agentic frameworks for multimodal generation.

2 Related Works

2.1 Inference-Time Scaling for Multimodal Generation

Recent years have witnessed significant progress in multimodal generation Wu et al. (2025); Cai et al. (2025); Brooks et al. (2024); He et al. (2025b); Saharia et al. (2022); Ramesh et al. (2022), and inference-time scaling has emerged as a promising strategy for performance enhancement. Early approaches primarily relied on simple prompt rewriting Hao et al. (2023) or random search Ma et al. (2025) to optimize generation. Other methods Fang et al. (2025); Jiang et al. (2025); Liao et al. (2025); Jiao et al. (2025); Kou et al. (2026) introduced Chain-of-Thought (CoT) Wei et al. (2022) reasoning to provide more guidance for multimodal generation. More advanced approaches Wang et al. (2025); Jiang et al. (2026b); Mondal et al. (2025); Li et al. (2025c); Zhuo et al. (2025); Li et al. (2025a); Guo et al. (2025) have adopted iterative refinement loops to progressively optimize the results. Recent studies Li et al. (2025b); Park et al. (2026); Wan et al. (2025); Kovalev et al. (2025); Venkatesh et al. (2025); Zhu et al. (2026) have also explored multi-agent systems. Some approaches Wan et al. (2025); Kovalev et al. (2025) leverage multi-agent collaboration and iterative optimization to enhance the generation process in complex tasks, yet are still limited to basic agent loops. Other studies Venkatesh et al. (2025); Zhu et al. (2026) focus on customized designs for specific downstream tasks, yet are often difficult to integrate with mainstream generative workflows. In contrast, GEMS adopts advanced agentic paradigms to address these limitations.

2.2 Agent Systems

Agent systems serve as autonomous frameworks that extend the reasoning and execution capabilities of LLMs through structured planning and interaction. Foundational works established agent loops Yao et al. (2022); Shinn et al. (2023); Wang et al. (2023); Madaan et al. (2023); Schick et al. (2023); He et al. (2025a) that enable models to alternate between reasoning and acting within a self-correcting cycle. Building on this, multi-agent systems Li et al. (2023); Hong et al. (2023); Wu et al. (2024); Chen et al. (2023) employ specialized roles that collaborate through communication protocols to tackle more intricate objectives. Furthermore, the integration of agent memory Packer et al. (2023); Chhikara et al. (2025); Xu et al. (2025); Long et al. (2025); Fu et al. (2026) enhances system performance in long-context and multi-turn interactions. More recently, agent skills Xia et al. (2026); Liang et al. (2026); Li et al. (2026); Jiang et al. (2026a) have further expanded the boundaries of agent systems, empowering them to execute complex tasks through domain-specific workflows. Building upon these capabilities, state-of-the-art agent systems such as Claude Code and OpenClaw have demonstrated remarkable capabilities in executing sophisticated, real-world operations, inspiring our adaptation of these agentic paradigms to multimodal generation.

3 Method

As shown in Figure 2, GEMS integrates Agent Loop, Agent Memory, and Agent Skill into a unified generation process. Agent Skill grounds the prompt in domain knowledge, while Agent Memory preserves visual evidence and distilled experiences that guide subsequent refinement within the Agent Loop. Together, they connect domain guidance with feedback from the evolving generation trajectory.

3.1 Agent Loop

Agent Loop serves as the backbone of GEMS, comprising several collaborative modules: Planner, Decomposer, Generator, Verifier, and Refiner.

Planner. The Planner, denoted as $\mathcal{F}_{plan}$, serves as the strategic entry point of the system. It first interacts with the Skill Manager to identify relevant expertise from the domain-specific repository $\mathcal{S}$ (Sec. 3.3) based on the user prompt U. This interaction retrieves a subset of triggered skills $\mathcal{S}_{trig} \subseteq \mathcal{S}$; if the task does not align with any specialized domain, $\mathcal{S}_{trig}$ remains empty. Leveraging the retrieved skills (if any), the Planner synthesizes an enhanced initial prompt P_1 designed to provide superior guidance for the generation process. Concurrently, it dispatches the original prompt U to the Decomposer to establish the foundational evaluation framework. The operation is defined as:

$$(P_1, U) = \mathcal{F}_{plan}(U, \mathcal{S}). \tag{1}$$

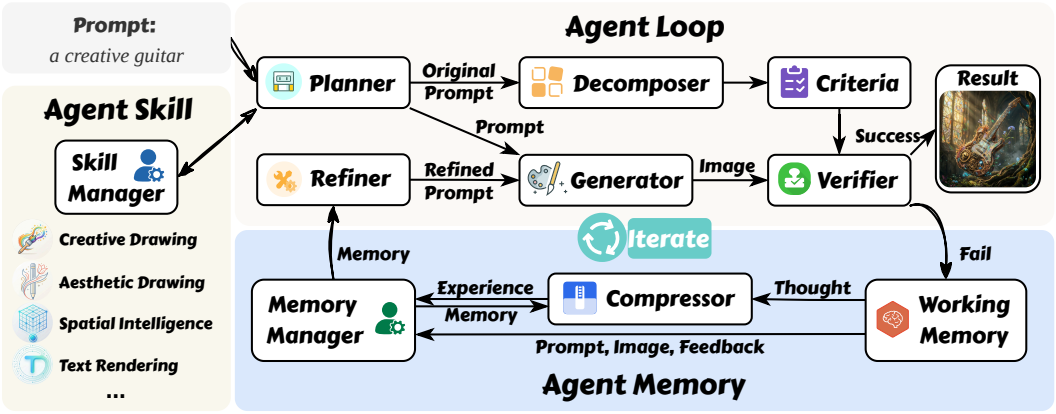


Figure 2: The system architecture of GEMS. The framework consists of three primary pillars: Agent Loop, Agent Memory, and Agent Skill. The user prompt is augmented with domain-specific expertise from Agent Skill, and then iteratively refined within the Agent Loop, with Agent Memory managing the historical context to guide the generation process.

Decomposer. To ensure fine-grained evaluation, the Decomposer $\mathcal{F}_{dec}$ partitions the user's original prompt U into a set of atomic visual requirements $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$. Each criterion c_j is formulated as a binary (yes/no) probe that represents an essential semantic or structural constraint:

$$\mathcal{C} = \mathcal{F}_{dec}(U). \tag{2}$$

Generator. The Generator $\mathcal{F}_{gen}$ is a model-agnostic module responsible for synthesizing the visual output. At each iteration i, it produces an image I_i based on the current optimized prompt P_i:

$$I_i = \mathcal{F}_{gen}(P_i). \tag{3}$$

Verifier. The Verifier $\mathcal{F}_{ver}$, powered by a Multimodal Large Language Model (MLLM), assesses the generated image I_i against the predefined atomic criteria set $\mathcal{C}$. It maps the visual and textual inputs to a binary feedback vector $V_i = \{v_{i,1}, \ldots, v_{i,n}\}$:

$$V_i = \mathcal{F}_{ver}(I_i, \mathcal{C}), \quad v_{i,j} \in \{0, 1\}. \tag{4}$$

The system then executes a conditional branch based on the result of V_i. If all criteria are met (i.e., $\forall j, v_{i,j} = 1$), the iterative loop terminates, and I_i is returned as the final output. If any criterion remains unsatisfied and the current iteration i is below the maximum limit N_{max}, the vector V_i is dispatched to the Refiner as diagnostic feedback. However, should the system reach N_{max} without satisfying all criteria, it performs a global evaluation over the optimization trajectory and returns the image I_{best} that fulfilled the maximum number of requirements:

$$I_{best} = \arg\max_{I_k} \sum_{j=1}^{n} v_{k,j}, \quad k \in \{1, \ldots, N_{max}\}. \tag{5}$$

Refiner. The Refiner $\mathcal{F}_{ref}$ facilitates prompt evolution by closing the feedback loop. At iteration i, it synthesizes a refined prompt P_{i+1} by analyzing the current state and historical context. Crucially, $\mathcal{M}_{i-1}$ represents the state of the Agent Memory at the conclusion of iteration $i-1$, which encapsulates the cumulative trajectory of preceding attempts. The Refiner integrates the current prompt P_i, the generated image I_i, the verification feedback V_i, and the internal reasoning T_i (reflecting the MLLM's thought process during refinement) with this historical state $\mathcal{M}_{i-1}$ to derive the next-turn prompt:

$$P_{i+1} = \mathcal{F}_{ref}(P_i, I_i, V_i, T_i, \mathcal{M}_{i-1}). \tag{6}$$

3.2 Agent Memory

Refining a generated image requires identifying unmet requirements while preserving what earlier attempts have already achieved. Relying only on the previous result or the best state can overlook

useful information from the generation trajectory, whereas retaining all reasoning traces introduces redundancy. We therefore maintain a persistent memory with a Hierarchical Compression strategy that treats generation evidence and reasoning differently. Specifically, the prompt P_i, generated image I_i, and verification feedback V_i are archived in their original form, allowing subsequent refinement to revisit earlier outputs and their assessed strengths and weaknesses. Reasoning traces T_i, which are often verbose and redundant Qu et al. (2025); Sui et al. (2025), are instead processed by a **Compressor** $\mathcal{F}_{comp}$ to distill concise experiences E_i that guide subsequent refinement:

$$E_i = \mathcal{F}_{comp}(P_i, I_i, V_i, T_i, \mathcal{M}_{i-1}). \tag{7}$$

The resulting memory state $\mathcal{M}_i$ is then updated as a sequence of these hybrid state tuples, ensuring that the system retains both factual anchors and strategic reflections:

$$\mathcal{M}_i = \{(P_1, I_1, V_1, E_1), \ldots, (P_i, I_i, V_i, E_i)\}. \tag{8}$$

This representation preserves generation evidence and adds concise strategic guidance, allowing the Refiner to draw on earlier successes and failures without accumulating full reasoning traces.

3.3 Agent Skill

Domain knowledge can improve generation for specialized tasks, but embedding it in separate workflows Venkatesh et al. (2025) makes it difficult to reuse across domains. Agent Skill organizes this knowledge as a repository of instructions for prompt enhancement, allowing different domains to share the same generation, verification, and refinement process. Before the iterative loop begins, the Planner selects relevant skills based on the user prompt and incorporates their guidance into the initial generation prompt. If no skill is relevant, the original prompt is used.

As illustrated in Figure 3, skill selection follows an on-demand loading and progressive exposure mechanism. The Planner first receives a manifest containing skill names and descriptions, then accesses the detailed instructions of selected skills. This keeps unselected instructions outside the reasoning context while making relevant domain guidance available for prompt enhancement. Each skill is specified in a markdown file, such as `SKILL.md`, so contributors can supply domain knowledge through a common interface without redesigning the agent workflow. The enhanced prompt then enters the shared refinement loop, where verification feedback and accumulated experiences guide subsequent updates.

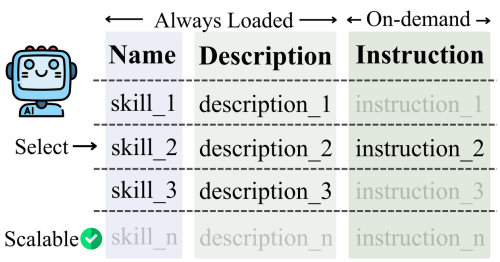


Figure 3: Architecture of the Agent Skill system, highlighting its scalable and on-demand nature.

4 Experiments

4.1 Experimental Setup

Implementation Details. To evaluate the effectiveness of GEMS, we conduct experiments with two distinct generative models: (1) **Z-Image-Turbo Cai et al. (2025)**: Z-Image is an efficient 6B model, and we further utilize its distilled version, Z-Image-Turbo, to prioritize inference efficiency. (2) **Qwen-Image-2512 Wu et al. (2025)**[1]: A representative 20B open-source model employed to verify the effectiveness of GEMS across different model architectures and parameter scales. We utilize Kimi K2.5 Team et al. (2026) as the MLLM backend. By default, the maximum number of iterations is set to 5. Four domain-oriented skills are enabled: Creative Drawing, Aesthetic Drawing, Text Rendering, and Spatial Intelligence. The maximum number of triggered skills is set to 1 for the main experiments. Further evaluation details and the skill construction procedure are provided in Appendix A.2 and Appendix A.3, respectively.

[1]Our evaluations indicate that Qwen-Image-2512 exhibits lower benchmark scores than Qwen-Image. This finding is consistent with results in other studies, such as the GLM-Image Technical Blog Z.ai (2026).

Table 1: Evaluation on Mainstream Tasks. For inference-time scaling methods, all results are evaluated by us, with the best and second-best performances highlighted in **bold** and underlined, and absolute score changes compared to the baseline indicated by ↑ and ↓, respectively. Other results are sourced from public data. The "Avg." column represents the mean of normalized scores, with OneIG-EN and OneIG-ZH pre-averaged as a single metric before final aggregation.

Model	Method	GenEval	GenEval2	DPG-Bench	OneIG-EN	OneIG-ZH	WISE	Avg.
Closed-Source Models								
Nano Banana	–	0.75	44.6	85.23	0.550	0.405	–	–
GPT-Image 1	–	0.84	–	85.15	0.533	0.474	0.80	–
Seedream 4 Seedream et al. (2025)	–	0.84	–	88.25	0.576	0.553	0.78	–
Open-Source Models								
Bagel Deng et al. (2025)	–	0.82	23.1	85.10	0.361	0.370	0.70	59.35
Z-Image Cai et al. (2025)	–	0.84	–	88.14	0.546	0.535	–	–
Qwen-Image Wu et al. (2025)	–	0.87	33.8	88.32	0.539	0.548	0.62	65.09
Inference-Time Scaling								
Z-Image-Turbo Cai et al. (2025)	Original	0.77 ↑0.00	31.0 ↑0.0	85.08 ↑0.00	0.526 ↑0.000	0.501 ↑0.000	0.57 ↑0.00	60.29
	Rewrite	0.77 ↑0.00	39.7 ↑8.7	84.48 ↓0.60	0.520 ↓0.006	0.469 ↓0.032	0.84 ↑0.27	66.93
	Promptist Hao et al. (2023)	0.76 ↓0.01	32.3 ↑1.3	65.69 ↓19.39	0.333 ↓0.193	0.327 ↓0.174	0.54 ↓0.03	52.20
	Search Ma et al. (2025)	0.81 ↑0.04	43.5 ↑12.5	85.47 ↑0.39	0.530 ↑0.004	0.506 ↑0.005	0.61 ↑0.04	64.55
	Maestro Wan et al. (2025)	0.82 ↑0.05	44.6 ↑13.6	85.29 ↑0.21	0.548 ↑0.022	0.519 ↑0.018	**0.85** ↑0.28	70.05
	CRAFT Kovalev et al. (2025)	0.80 ↑0.03	62.4 ↑31.4	85.29 ↑0.21	**0.582** ↑0.056	0.542 ↑0.041	0.78 ↑0.21	72.38
	GEMS (Ours)	**0.86** ↑0.09	**63.5** ↑32.5	**86.01** ↑0.93	0.569 ↑0.043	**0.552** ↑0.051	0.81 ↑0.24	**74.51**
Qwen-Image-2512 Wu et al. (2025)	Original	0.66 ↑0.00	29.0 ↑0.0	84.69 ↑0.00	0.487 ↑0.000	0.489 ↑0.000	0.59 ↑0.00	57.50
	Rewrite	0.68 ↑0.02	39.8 ↑10.8	83.49 ↓1.20	0.465 ↓0.022	0.435 ↓0.054	0.81 ↑0.22	63.46
	Promptist Hao et al. (2023)	0.66 ↑0.00	27.4 ↓1.6	64.02 ↓20.67	0.288 ↓0.199	0.307 ↓0.182	0.57 ↓0.02	48.83
	Search Ma et al. (2025)	0.74 ↑0.08	42.1 ↑13.1	**86.17** ↑1.48	0.502 ↑0.015	0.497 ↑0.008	0.67 ↑0.08	63.84
	Maestro Wan et al. (2025)	0.76 ↑0.10	48.0 ↑19.0	84.07 ↓0.62	0.494 ↑0.007	0.489 ↑0.000	**0.84** ↑0.25	68.24
	CRAFT Kovalev et al. (2025)	**0.79** ↑0.13	66.3 ↑37.3	85.87 ↑1.18	0.533 ↑0.046	0.518 ↑0.029	0.79 ↑0.20	72.54
	GEMS (Ours)	**0.79** ↑0.13	**70.4** ↑41.4	85.59 ↑0.90	**0.542** ↑0.055	**0.532** ↑0.043	0.80 ↑0.21	**73.74**

Benchmarks and Baselines We divide the evaluation into two groups with complementary purposes. We use five mainstream benchmarks, including GenEval Ghosh et al. (2023), GenEval2 Kamath et al. (2025), DPG-Bench Hu et al. (2024), OneIG Chang et al. (2025), and WISE Niu et al. (2025), to assess general applicability, including whether the same skill repository benefits generation tasks that were not targeted during skill construction. We use four downstream tasks, including LongText-Bench Geng et al. (2025), SpatialGenEval Wang et al. (2026), CREA Venkatesh et al. (2025), and ArtiMuse Cao et al. (2025b), to assess whether domain knowledge provides additional gains in the capabilities for which the skills were designed. We use the same fixed skill repository in both groups; Figure 6 isolates its contribution through comparisons with the variant without Agent Skill. Our baselines consist of strong closed-source and open-source generative models, as well as inference-time scaling systems. To ensure a fair comparison under similar computational budgets, we specifically set the parallelism factor for Search Ma et al. (2025) to 5, and limit the maximum number of iterations for Maestro Wan et al. (2025) and CRAFT Kovalev et al. (2025) to 3 and 5, respectively. A detailed comparison of the computational costs versus performance gains for the various methods is illustrated in Figure 8. Further details are provided in Appendix A.4 and Appendix A.5.

4.2 Main Results

Tables 1 and 2 present the experimental results across mainstream and downstream tasks, respectively. On mainstream tasks, GEMS, leveraging Z-Image-Turbo, achieves consistent performance gains with an average increase of 14.22 in normalized scores, outperforming prior inference-time scaling baselines. Further validation on Qwen-Image-2512 confirms the generalizability and effectiveness of our approach across different foundational architectures.

On downstream tasks, GEMS demonstrates an even more pronounced advantage, yielding an average improvement of 14.03 in normalized scores with Z-Image-Turbo, and significantly surpassing the best-performing inference-time scaling baseline (+8.92). Notably, we observe that several baseline methods involving prompt rewriting, such as Rewrite and Promptist Hao et al. (2023), exhibit significant performance degradation in certain tasks, particularly in text rendering. This decline stems from the fact that general-purpose rewriting strategies often lack domain-specific constraints, frequently compromising strict textual information during the optimization process. In contrast, GEMS incorporates specialized skills to provide targeted guidance for optimization, resulting in consistent and substantial performance enhancements even in highly specialized domains.

Table 2: Evaluation on Downstream Tasks. The "Avg." column represents the mean of normalized scores, with LongText-EN and LongText-ZH pre-averaged as a single metric before final aggregation.

Model	Method	LongText-EN	LongText-ZH	SpatialGenEval	CREA	ArtiMuse	Avg.
Closed-Source Models							
Nano Banana	–	–	–	61.7	–	–	–
GPT-Image 1	–	0.956	0.619	60.5	–	–	–
Seedream 4 Seedream et al. (2025)	–	–	–	62.7	–	–	–
Open-Source Models							
Bagel Deng et al. (2025)	–	0.373	0.310	57.0	–	–	–
Z-Image Cai et al. (2025)	–	0.935	0.936	–	–	–	–
Qwen-Image Wu et al. (2025)	–	0.943	0.946	60.6	–	–	–
Inference-Time Scaling							
Z-Image-Turbo Cai et al. (2025)	Original	0.912 ↑0.000	0.932 ↑0.000	58.7 ↑0.0	11.84 ↑0.00	43.27 ↑0.00	58.41
	Rewrite	0.571 ↓0.341	0.495 ↓0.437	59.6 ↑0.9	16.75 ↑4.91	57.86 ↑14.59	56.65
	Promptist Hao et al. (2023)	0.034 ↓0.878	0.070 ↓0.862	48.3 ↓10.4	14.00 ↑2.16	43.74 ↑0.47	35.98
	Search Ma et al. (2025)	0.918 ↑0.006	0.937 ↑0.005	59.3 ↑0.6	12.98 ↑1.14	44.21 ↑0.94	59.88
	Maestro Wan et al. (2025)	0.877 ↓0.035	0.807 ↓0.125	60.3 ↑1.6	15.81 ↑3.97	56.86 ↑13.59	63.52
	CRAFT Kovalev et al. (2025)	0.951 ↑0.039	0.760 ↓0.172	60.6 ↑1.9	13.63 ↑1.79	54.95 ↑11.68	61.63
	GEMS (Ours)	**0.952** ↑0.040	**0.940** ↑0.008	**61.4** ↑2.7	**22.55** ↑10.71	**58.58** ↑15.31	**72.44**
Qwen-Image-2512 Wu et al. (2025)	Original	0.854 ↑0.000	0.892 ↑0.000	60.0 ↑0.0	17.45 ↑0.00	59.94 ↑0.00	66.35
	Rewrite	0.563 ↓0.291	0.468 ↓0.424	60.3 ↑0.3	18.12 ↑0.67	62.23 ↑2.29	58.62
	Promptist Hao et al. (2023)	0.030 ↓0.824	0.058 ↓0.834	48.7 ↓11.3	18.35 ↑0.90	60.65 ↑0.71	43.73
	Search Ma et al. (2025)	0.883 ↑0.029	0.907 ↑0.015	60.6 ↑0.6	18.71 ↑1.26	59.96 ↑0.02	68.11
	Maestro Wan et al. (2025)	0.833 ↓0.021	0.792 ↓0.100	61.0 ↑1.0	17.78 ↑0.33	61.62 ↑1.68	65.78
	CRAFT Kovalev et al. (2025)	0.897 ↑0.043	0.787 ↓0.105	61.6 ↑1.6	15.61 ↓1.84	59.78 ↓0.16	64.40
	GEMS (Ours)	**0.913** ↑0.059	**0.931** ↑0.039	**62.1** ↑2.1	**24.01** ↑6.56	**62.89** ↑2.95	**74.31**

4.3 Ablation and Discussion

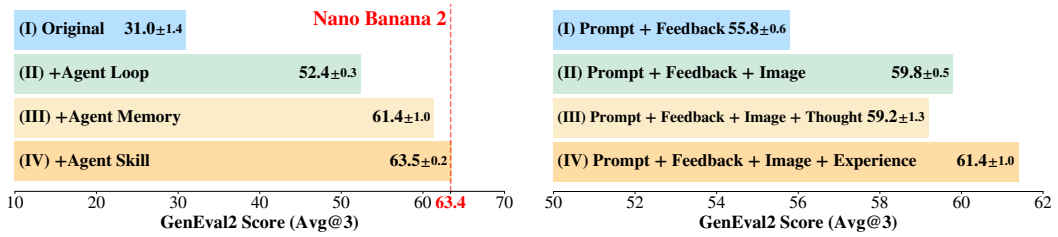


Figure 4: Ablation study on GenEval2 with Z-Image-Turbo. Results are reported as mean ± standard deviation over three runs. (Left) Performance gains contributed by individual components, including Agent Loop, Agent Memory, and Agent Skill. (Right) Detailed analysis of Agent Memory.

Overall Ablation Study. We ablate GEMS on GenEval2, selected due to its status as a challenging and unsaturated benchmark. To improve the reliability of the key ablation results, we conduct three independent runs on GenEval2 using Z-Image-Turbo and report the mean scores and standard deviations. As shown in Figure 4(left), the sequential integration of the Agent Loop, Agent Memory, and Agent Skill yields substantial performance gains. Specifically, the basic Agent Loop improves the score from 31.0 to 52.4, while the addition of Agent Memory and Agent Skill contributes further increases of 9.0 and 2.1 points, respectively, culminating in a final score of 63.5. Notably, GEMS enables the lightweight generator to match the performance of Nano Banana 2 on GenEval2. This demonstrates that GEMS effectively unlocks the potential of foundational models, allowing them to transcend inherent capacity limits through agentic reasoning and domain-specific expertise.

Analysis of Agent Loop As shown in Figure 4(left), the Agent Loop itself provides a substantial performance boost. A primary factor is the inherent stochasticity of the image generation process; within an iterative framework, as long as a single iteration produces a valid output, the Verifier can identify it as a success. In this sense, the loop partially functions like a Random Search Ma et al. (2025) strategy by providing multiple "shots" at the target.

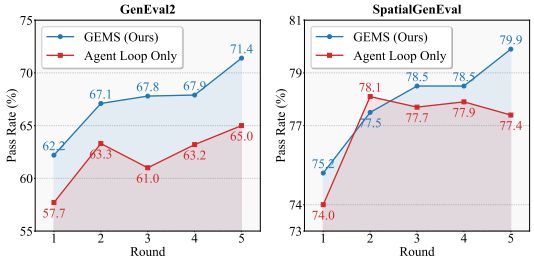


Figure 5: Average passed criteria over iterations.

However, GEMS goes beyond mere repetition. To demonstrate that the prompt quality actually improves over time, we analyzed the average number of passed criteria across iterations on the most

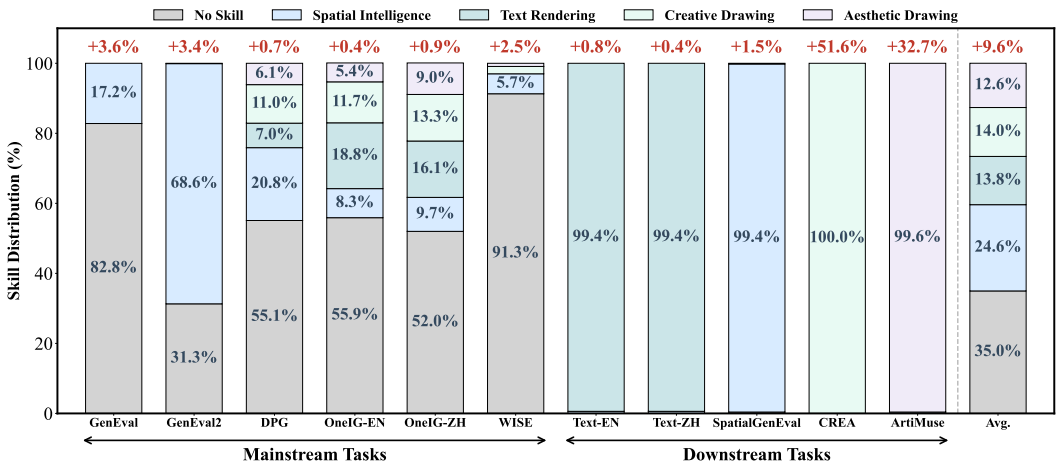


Figure 6: Skill distribution across tasks and relative performance improvement. Each stacked bar illustrates the percentage composition of skills triggered within the respective benchmarks. The red numerical labels (e.g., +3.6%) above the bars denote the relative performance gain achieved compared to the version without Agent Skill.

challenging benchmarks: GenEval2 and SpatialGenEval. As illustrated in Figure 5, while a basic Agent Loop Only approach shows some initial gains, its performance tends to fluctuate (e.g., in SpatialGenEval). In contrast, GEMS starts from a higher initial baseline and exhibits a consistent upward trajectory in success rate. For instance, on GenEval2, it progressively climbs from 62.2% to 71.4%, widening the margin over the baseline as rounds progress. This trend indicates that the Refiner is not merely generating random variations, but is actively performing directed optimization based on feedback. This ensures that GEMS fundamentally outperforms naive iterative methods.

Analysis of Agent Memory We further investigate the impact of different Agent Memory configurations, as illustrated in Figure 4(right). Initially, incorporating only historical prompts and their corresponding feedback leads to a 3.4 point improvement. Further including the generated images into the memory pool contributes an additional 4.0 points, suggesting that richer multimodal context provides more robust guidance for the refinement process.

However, we find that more information is not always beneficial. When we attempt to include the full thought (CoT used to generate the corresponding prompt) into the memory, it does not result in performance gains. Raw reasoning logs can contain substantial redundancy Qu et al. (2025); Sui et al. (2025), which may distract the Refiner. We therefore utilize the Compressor to summarize previous attempts into concise "Experiences", providing strategic guidance without accumulating full reasoning traces. Using these experiences yields a score of 61.4, improving by 2.2 points over retaining full thought, supporting the use of concise experiences to guide subsequent refinement.

Analysis of Agent Skill. We first examine the impact of skills on the downstream tasks for which they were designed. Figure 6 reports skill activation frequencies and compares GEMS with the variant without Agent Skill. The system selects relevant domain guidance, with Spatial Intelligence predominantly activated for SpatialGenEval and Creative Drawing for CREA. Compared with the variant without Agent Skill, GEMS achieves relative improvements of 51.6% on CREA, and 32.7% on ArtiMuse. Figure 7 further illustrates how Creative Drawing and Aesthetic Drawing improve visual appeal and composition. Appendix C further examines the joint use of more than one skills.

We also assess whether the same fixed skill repository generalizes to mainstream tasks that were not targeted during skill construction. Figure 6 shows selective activation on mainstream benchmarks, with spatial requirements frequently triggering Spatial Intelligence. Compared with the variant without Agent Skill, GEMS achieves relative improvements of 3.6% on GenEval and 3.4% on GenEval2, supporting the reuse of domain guidance beyond its intended downstream applications. The detailed GenEval results in Appendix B, Table 4, further show that the "Position" category exhibits the largest gain of +0.34 with Z-Image-Turbo. These results highlight the value of reusable skill guidance in improving performance for general generation tasks.



Figure 7: Qualitative Comparison of Agent Skills. GEMS autonomously triggers Aesthetic Drawing (left) and Creative Drawing (right) to enhance artistic expression.

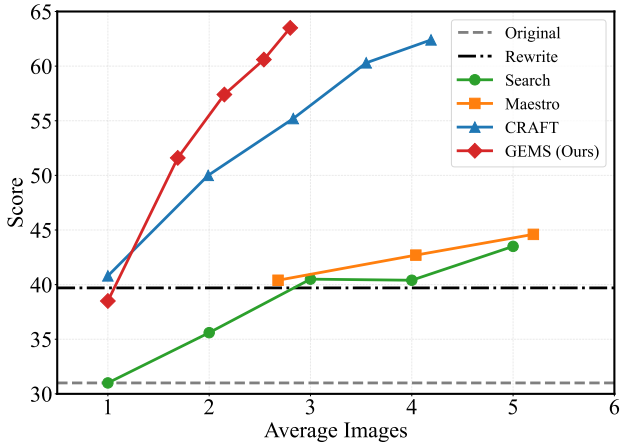


Figure 8: Trade-off between Efficiency and Performance. Comparison of inference-time scaling methods on GenEval2 using Z-Image-Turbo. GEMS (red line) achieves superior performance with fewer average images generated.

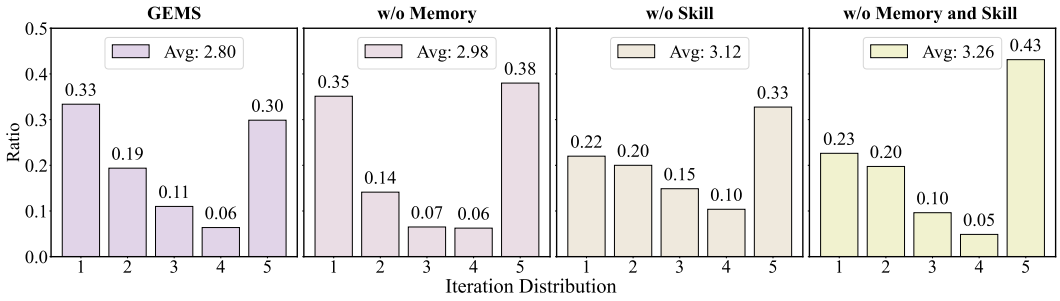


Figure 9: Iteration distribution across GEMS ablation variants. Incorporating Memory and Skill shifts the distribution toward earlier termination and reduces the average number of iterations.

Trade-off between Efficiency and Performance. We also investigate the scaling behavior of various inference-time methods, focusing on the trade-off between computational efficiency and generative quality. As illustrated in Figure 8, we evaluate these methods using Z-Image-Turbo on GenEval2, tracking the average number of generated images against the resulting scores, which serves as a uniform proxy across frameworks since image synthesis often dominates the cost. Due to the early stopping mechanism, GEMS delivers superior performance while maintaining significantly lower overhead. For instance, at an average of approximately three images per task, GEMS substantially outperforms other inference-time scaling methods. Figure 9 further shows that incorporating Memory and Skill shifts the iteration distribution toward earlier termination, reducing the average number of iterations from 3.26 to 2.80. This complements the performance comparison by showing that the full framework reaches its stopping criteria with fewer generation steps. Appendix C.3 provides a broader cost analysis, including MLLM calls, token usage, estimated API costs, and total evaluation time.

5 Conclusion

We present GEMS, an agent-native multimodal generation framework that connects reusable domain guidance with accumulated visual experience in a unified refinement process. By preserving generation artifacts, distilling reasoning into experiences, and incorporating domain knowledge through a shared prompt interface, GEMS supports complex instruction following and specialized downstream tasks within the same workflow. With Z-Image-Turbo, GEMS achieves average score gains of 14.22 across five mainstream benchmarks and 14.03 across four downstream tasks, matching Nano Banana 2 on GenEval2 with a 6B generator. These results demonstrate the value of jointly organizing domain knowledge and generation history, offering a practical direction for agentic multimodal generation.

AI USE STATEMENT

AI tools were used to assist with language polishing. The authors reviewed the resulting text and take responsibility for the content of this paper. Also, AI assistance in skill construction is described in Appendix A.3.

REPRODUCIBILITY STATEMENT

The appendix provides the agent prompts, evaluation settings, and skill construction procedure to support reproducibility. We also provide our implementation details and skill instructions through the anonymous link in abstract.

REFERENCES

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

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.

Shuo Cao, Jiayang Li, Xiaohui Li, Yuandong Pu, Kaiwen Zhu, Yuanting Gao, Siqi Luo, Yi Xin, Qi Qin, Yu Zhou, et al. Unipercept: Towards unified perceptual-level image understanding across aesthetics, quality, structure, and texture. *arXiv preprint arXiv:2512.21675*, 2025a.

Shuo Cao, Nan Ma, Jiayang Li, Xiaohui Li, Lihao Shao, Kaiwen Zhu, Yu Zhou, Yuandong Pu, Jiarui Wu, Jiaquan Wang, et al. Artimuse: Fine-grained image aesthetics assessment with joint scoring and expert-level understanding. *arXiv preprint arXiv:2507.14533*, 2025b.

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

Guanjie Chen, Shirui Huang, Kai Liu, Jianchen Zhu, Xiaoye Qu, Peng Chen, Yu Cheng, and Yifu Sun. Flash-dmd: Towards high-fidelity few-step image generation with efficient distillation and joint reinforcement learning. *arXiv preprint arXiv:2511.20549*, 2025.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations*, 2023.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

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.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Rongyao Fang, Chengqi Duan, Kun Wang, Linjiang Huang, Hao Li, Shilin Yan, Hao Tian, Xingyu Zeng, Rui Zhao, Jifeng Dai, et al. Got: Unleashing reasoning capability of multimodal large language model for visual generation and editing. *arXiv preprint arXiv:2503.10639*, 2025.

Muxin Fu, Guibin Zhang, Xiangyuan Xue, Yafu Li, Zefeng He, Siyuan Huang, Xiaoye Qu, Yu Cheng, and Yang Yang. Latentmem: Customizing latent memory for multi-agent systems. *arXiv preprint arXiv:2602.03036*, 2026.

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.

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

Ziyu Guo, Renrui Zhang, Hongyu Li, Manyuan Zhang, Xinyan Chen, Sifan Wang, Yan Feng, Peng Pei, and Pheng-Ann Heng. Thinking-while-generating: Interleaving textual reasoning throughout visual generation. *arXiv preprint arXiv:2511.16671*, 2025.

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.

Zefeng He, Xiaoye Qu, Yafu Li, Siyuan Huang, Daizong Liu, and Yu Cheng. Framethinker: Learning to think with long videos via multi-turn frame spotlighting. *arXiv preprint arXiv:2509.24304*, 2025a.

Zefeng He, Xiaoye Qu, Yafu Li, Tong Zhu, Siyuan Huang, and Yu Cheng. Diffthinker: Towards generative multimodal reasoning with diffusion models. *arXiv preprint arXiv:2512.24165*, 2025b.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *The twelfth international conference on learning representations*, 2023.

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.

Dongzhi Jiang, Ziyu Guo, Renrui Zhang, Zhuofan Zong, Hao Li, Le Zhuo, Shilin Yan, Pheng-Ann Heng, and Hongsheng Li. T2i-r1: Reinforcing image generation with collaborative semantic-level and token-level cot. *arXiv preprint arXiv:2505.00703*, 2025.

Guanyu Jiang, Zhaochen Su, Xiaoye Qu, et al. Xskill: Continual learning from experience and skills in multimodal agents. *arXiv preprint arXiv:2603.12056*, 2026a.

Kaixun Jiang, Yuzheng Wang, Junjie Zhou, Pandeng Li, Zhihang Liu, Chen-Wei Xie, Zhaoyu Chen, Yun Zheng, and Wenqiang Zhang. Genagent: Scaling text-to-image generation via agentic multimodal reasoning. *arXiv preprint arXiv:2601.18543*, 2026b.

Siyu Jiao, Yiheng Lin, Yujie Zhong, Qi She, Wei Zhou, Xiaohan Lan, Zilong Huang, Fei Yu, Yingchen Yu, Yunqing Zhao, et al. Thinkgen: Generalized thinking for visual generation. *arXiv preprint arXiv:2512.23568*, 2025.

Amita Kamath, Kai-Wei Chang, Ranjay Krishna, Luke Zettlemoyer, Yushi Hu, and Marjan Ghazvininejad. Geneval 2: Addressing benchmark drift in text-to-image evaluation. *arXiv preprint arXiv:2512.16853*, 2025.

Siqi Kou, Jiachun Jin, Zetong Zhou, Ye Ma, Yugang Wang, Quan Chen, Peng Jiang, Xiao Yang, Jun Zhu, Kai Yu, et al. Think-then-generate: Reasoning-aware text-to-image diffusion with llm encoders. *arXiv preprint arXiv:2601.10332*, 2026.

V Kovalev, A Kuvshinov, A Buzovkin, D Pokidov, and D Timonin. Craft: Continuous reasoning and agentic feedback tuning for multimodal text-to-image generation. *arXiv preprint arXiv:2512.20362*, 2025.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in neural information processing systems*, 36:51991–52008, 2023.

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*, 2025a.

Mingcheng Li, Xiaolu Hou, Ziyang Liu, Dingkang Yang, Ziyun Qian, Jiawei Chen, Jinjie Wei, Yue Jiang, Qingyao Xu, and Lihua Zhang. Mccd: Multi-agent collaboration-based compositional diffusion for complex text-to-image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13263–13272, 2025b.

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, 2025c.

Xiangyi Li, Wenbo Chen, Yimin Liu, Shenghan Zheng, Xiaokun Chen, Yifeng He, Yubo Li, Bingran You, Haotian Shen, Jiankai Sun, et al. Skillsbench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*, 2026.

Yuan Liang, Ruobin Zhong, Haoming Xu, Chen Jiang, Yi Zhong, Runnan Fang, Jia-Chen Gu, Shumin Deng, Yunzhi Yao, Mengru Wang, et al. Skillnet: Create, evaluate, and connect ai skills. *arXiv preprint arXiv:2603.04448*, 2026.

Jiaqi Liao, Zhengyuan Yang, Linjie Li, Dianqi Li, Kevin Lin, Yu Cheng, and Lijuan Wang. Imagegen-cot: Enhancing text-to-image in-context learning with chain-of-thought reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17214–17223, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. *arXiv preprint arXiv:2508.09736*, 2025.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Anindya Mondal, Ayan Banerjee, Sauradip Nag, Josep LladÃġs, Xiatian Zhu, and Anjan Dutta. Countloop: Training-free high-instance image generation via iterative agent guidance. *arXiv preprint arXiv:2508.16644*, 2025.

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.

Charles Packer, Vivian Fang, Shishir_G Patil, Kevin Lin, Sarah Wooders, and Joseph_E Gonzalez. Memgpt: towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Sangha Park, Eunji Kim, Yeongtak Oh, Jooyoung Choi, and Sungroh Yoon. Guiding what not to generate: Automated negative prompting for text-image alignment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6664–6675, 2026.

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

Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, et al. A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond. *arXiv preprint arXiv:2503.21614*, 2025.

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.

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

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.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

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.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, and Xia Hu. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

Kavana Venkatesh, Connor Dunlop, and Pinar Yanardag. Crea: A collaborative multi-agent framework for creative image editing and generation. *arXiv preprint arXiv:2504.05306*, 2025.

Xingchen Wan, Han Zhou, Ruoxi Sun, Hootan Nakhost, Ke Jiang, Rajarishi Sinha, and Sercan Ö Arık. Maestro: Self-improving text-to-image generation via agent orchestration. *arXiv preprint arXiv:2509.10704*, 2025.

Kaishen Wang, Ruibo Chen, Tong Zheng, and Heng Huang. Imagent: A unified multimodal agent framework for test-time scalable image generation. *arXiv preprint arXiv:2511.11483*, 2025.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 2609–2634, 2023.

Zengbin Wang, Xuecai Hu, Yong Wang, Feng Xiong, Man Zhang, and Xiangxiang Chu. Everything in its place: Benchmarking spatial intelligence of text-to-image models. *arXiv preprint arXiv:2601.20354*, 2026.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Li Fei-Fei, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

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.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First conference on language modeling*, 2024.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, et al. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

Z.ai. Glm-image: Auto-regressive for dense-knowledge and high-fidelity image generation, 2026. URL `https://z.ai/blog/glm-image`.

Dawei Zhu, Rui Meng, Yale Song, Xiyu Wei, Sujian Li, Tomas Pfister, and Jinsung Yoon. Paperbanana: Automating academic illustration for ai scientists. *arXiv preprint arXiv:2601.23265*, 2026.

Le Zhuo, Liangbing Zhao, Sayak Paul, Yue Liao, Renrui Zhang, Yi Xin, Peng Gao, Mohamed Elhoseiny, and Hongsheng Li. From reflection to perfection: Scaling inference-time optimization for text-to-image diffusion models via reflection tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15329–15339, 2025.

APPENDIX

A Experiment Details **16**

A.1 Prompt 16

A.2 Evaluation Details 18

A.3 Skill Construction 18

A.4 Benchmark Details 18

A.5 Baseline Details 19

B Additional Results **19**

B.1 Analysis of Performance 19

B.2 Results by Benchmark Category 20

C Additional Experiments **24**

C.1 Combining Multiple Skills 24

C.2 Alternative MLLM Backend 24

C.3 Inference Cost 24

D Qualitative Results **24**

E Limitations and Future Work **26**

Prompt for *Planner* (Skill Selection)

```
You are a strategic Skill Router. Your goal is to determine if the user's request genuinely
requires a specialized skill or if it can be handled by standard generation.

### Available Skills:
{manifest}
### User Request:
{user_prompt}

### Evaluation Criteria:
1. **Relevance**: Does the request explicitly match the 'DESCRIPTION' of a skill?
2. **Added Value**: Does using this skill provide significant benefits (e.g., specific artistic
styles, complex logic, or reference handling) that standard generation lacks?
3. **Default to NONE**: If the request is simple, generic, or does not strongly align with any
skill, you MUST choose 'NONE'.

### Response Requirement:
- If a skill is a strong match, respond ONLY with the SKILL_ID.
- Otherwise, respond ONLY with 'NONE'.
Do not provide any explanation, preamble, or punctuation.
```

Figure 10: Prompt for Planner (Skill Selection).

Prompt for *Planner* (Skill Application)

```
Based on the following skill instructions, enhance the user's prompt.
### Skill Instructions:
{skill_info['instructions']}

### Original Prompt: {original_prompt}

Return ONLY the final enhanced prompt text.
```

Figure 11: Prompt for Planner (Skill Application).

A Experiment Details

A.1 Prompt

The detailed prompts for the LLM-based modules (Planner, Decomposer, Verifier, Refiner, and Compressor) are presented in Figures 10 through 15. In contrast, the Skill Manager and Memory Manager are implemented as programmatic modules (e.g., Python class) rather than LLM agents.

Prompt for *Decomposer*

Analyze the user's image generation prompt. Break it down into specific visual requirements. For each requirement, write a question that can be answered with a simple 'yes' or 'no'. The questions should verify if the requirement is met in an image. YOU MUST RESPOND ONLY WITH A JSON ARRAY OF STRINGS. Example format: ["Is there a cat?", "Is the cat black?", "Is it sitting on a rug?"]

Figure 12: Prompt for Decomposer.

Prompt for *Verifier*

Image: <image>
Answer the following question with only 'yes' or 'no' based on the provided image:

Figure 13: Prompt for Verifier.

Prompt for *Refiner*

Task: Refine the image generation prompt based on previous failed attempts and accumulated experiences.
Original Intent: {original_prompt}

--- ATTEMPT HISTORY ---
{history_log}
--- ANALYSIS ---
Review the history above. Rewrite a new, comprehensive prompt. This prompt must:
1. Explicitly reinforce the requirements that failed in the latest attempt.
2. Maintain and protect the requirements that were successfully met in previous rounds to avoid regressions.
3. Adopt the strategies suggested in the 'Experience' section.
4. Use clear, non-conflicting descriptive language.

Return ONLY the prompt text itself. Do not include any conversational filler, introductory remarks, or prefixes like 'New prompt:' or 'Refined prompt:'.

Figure 14: Prompt for Refiner.

Prompt for *Compressor*

Task: Summarize the experience of the current image generation attempt.
--- CURRENT ATTEMPT ---
Prompt used: {current_prompt}
Passed requirements: {passed}
Failed requirements: {failed}
Reasoning/Thought before generation: {current_thought}
Image: <image>
--- PREVIOUS EXPERIENCES ---
{previous_experiences}
--- ANALYSIS ---
Based on the provided image, the verification results, your previous thought process, and historical experiences, write a concise summary (experience) of what worked, what failed, and what strategy should be adopted in the next attempt. Keep it under 100 words. Do not include introductory phrases.

Figure 15: Prompt for Compressor.

A.2 Evaluation Details

We conduct our evaluations following the official settings of Z-Image Cai et al. (2025) and Qwen-Image Wu et al. (2025). For a comprehensive comparison, the configuration of the state-of-the-art Nano Banana 2 is also presented. Detailed specifications for these models are summarized in Table 3.

Table 3: Evaluation Details.

Model	Resolution	DiT Forwards
Z-Image-Turbo	1024×1024	8
Qwen-Image-2512	1328×1328	50
Nano Banana 2	1024×1024	–

A.3 Skill Construction

We construct four domain-oriented skills: Creative Drawing, Aesthetic Drawing, Text Rendering, and Spatial Intelligence. Each skill contains a description for routing and detailed instructions for prompt enhancement. We first define high-level human design principles and collect reference materials for the corresponding domain. Gemini-3-Pro is then used to draft the skill instructions, which are subsequently reviewed by humans. The complete instructions for all four skills are included as `SKILL.md` files in the anonymous link in abstract.

The skills encode domain knowledge for the downstream capabilities studied in this work. After construction, the repository is fixed and is not tuned or modified based on benchmark feedback. We do not use the prompt distributions, category definitions, evaluation protocols, or evaluators of the five mainstream benchmarks to construct the skills. Accordingly, downstream evaluation measures the benefit of knowledge in its target domains, whereas mainstream evaluation measures the reuse of the same repository on non-targeted tasks. This distinction concerns the targets of skill construction; it does not imply that the downstream skills were developed without reference to domain-specific evaluation concepts.

At inference time, the Planner first considers the skill names and descriptions and loads the detailed instructions only for a selected skill. It uses these instructions to enhance the original prompt before iterative generation. If no skill is relevant, it returns `NONE` and the system proceeds with the original prompt. The main experiments permit at most one skill per prompt; the additional experiment in Appendix C also considers a two-skill cap.

Skill content overview. The four skills provide reusable guidance for creative drawing, aesthetic quality, text rendering, and spatial composition. Their complete instructions are available through the anonymous code link.

Execution settings. The main experiments use Kimi K2.5 as the MLLM backend, a maximum of five iterations, and at most one selected skill per prompt. The generator resolutions and DiT forward counts are listed in Table 3, and the agent prompts are provided in Appendix A.1. Skill loading refers to supplying the selected instructions to the MLLM context. It does not require that the corresponding files be read from disk only at that moment. The two-skill setting is an additional experiment described in Appendix C.1.

A.4 Benchmark Details

The four downstream tasks (LongText-Bench, SpatialGenEval, CREA, and ArtiMuse) primarily focus on evaluating the model's capabilities in text rendering, spatial intelligence, creative drawing, and aesthetic drawing, respectively. For ArtiMuse test, we employ the ArtiMuse Cao et al. (2025b) model as our dedicated scoring metric to evaluate the aesthetic quality of the generated images. Following the experimental protocol in Unipercept Cao et al. (2025a), we adopt prompts from the GenEval benchmark. To ensure the generative systems are evaluated at their aesthetic potential, we prepend the prefix "Aesthetic Drawing: " to each prompt, thereby steering the models toward generating results with higher visual appeal for ArtiMuse to assess.Regarding CREA Venkatesh et al. (2025), since it is not fully open-sourced, we evaluate on half of its publicly available data. The evaluation subset, configuration details are provided in the implementation available through the anonymous code link. For all other benchmarks, we strictly adhere to the official scripts.

A.5 Baseline Details

In this section, we provide details of the inference-time scaling baselines compared in our study.

Rewrite. The Rewrite baseline employs the MLLM to enhance initial user prompts following established best practices from the Google Cloud text-to-image prompt guide.

Promptist Hao et al. (2023). The Promptist baseline utilizes a fine-tuned language model to automatically rephrase user prompts. It generates optimized refinements via beam search, aiming to better align user intent with the capabilities of text-to-image models.

Random Search Ma et al. (2025). The Random Search baseline generates multiple images and selects the best based on scores from a verifier. In our implementation, we utilize Kimi as verifier.

Maestro Wan et al. (2025). Maestro is a self-improving text-to-image framework that utilizes iterative generation, producing two images in each round and featuring a pairwise comparison mechanism to evolve. Since the official code is not publicly available, we re-implemented the framework as a baseline based on the original paper, specifically incorporating its signature pairwise comparison logic.

CRAFT Kovalev et al. (2025). CRAFT is an iterative framework that refines prompts based on MLLM-based feedback. It evaluates images against generated visual questions and performs targeted prompt updates to address identified failures, without employing specialized memory management.

Evaluation Protocol for Agent Skill. Our main comparisons assess complete frameworks with their respective capabilities. GEMS uses the same fixed skill repository across all evaluated tasks. The Planner automatically selects relevant instructions based on the input prompt and incorporates them into the initial generation prompt before iterative refinement, without manual skill assignment to individual benchmarks or prompts. Baselines such as Rewrite, Maestro, and CRAFT retain their evaluated workflows without the GEMS Planner or skill repository, since these components form part of the proposed method. Baseline methods would also be expected to benefit from relevant skill instructions manually selected for each prompt or benchmark. Such a setup, however, introduces human intervention in knowledge selection and therefore differs from the autonomous evaluation setting considered here. Alternatively, the same automatic Planner and fixed skill repository from GEMS could be placed before a baseline's refinement workflow. This would transfer GEMS's knowledge integration capability to the baseline, producing an augmented configuration that combines Agent Skill with the baseline's refinement mechanism. Our main comparisons therefore retain the respective framework configurations to evaluate their overall capabilities.

Positioning of GEMS. A main contribution of GEMS is to integrate reusable domain knowledge into a shared generation workflow and demonstrate the practical benefits of this design across tasks. Such knowledge can substantially benefit specialized downstream tasks and also improve mainstream generation when relevant guidance is available. Specialized systems Zhu et al. (2026); Venkatesh et al. (2025) incorporate domain knowledge into workflows tailored to particular applications, limiting its direct reuse in a general generation pipeline. Meanwhile, general-purpose baselines such as Rewrite, Maestro, and CRAFT perform prompt refinement, but their evaluated workflows lack an explicit interface for automatically selecting and incorporating instructions from a reusable domain repository. Agent Skill provides a simple interface that makes this guidance available within a common generation, verification, and refinement process. Automatic selection serves as a practical means of accessing relevant expertise; our emphasis is on the integration and use of domain guidance, rather than the superiority of a particular routing algorithm.

B Additional Results

B.1 Analysis of Performance

Performance and efficiency. Note that strong baselines such as CRAFT also achieve competitive performance on GenEval2, scoring 62.4 compared with 63.5 for GEMS using Z-Image-Turbo.

However, these results require different numbers of generation attempts. As shown in Figure 8, with both methods limited to five iterations, CRAFT generates 4.19 images per prompt on average, whereas GEMS generates 2.80, requiring 33.2% fewer images. Reducing CRAFT's maximum iterations to three or four yields average image counts of 2.83 and 3.55, respectively, with scores of 55.2 and 60.3. GEMS thus achieves higher scores with fewer generation attempts, including an 8.3- point advantage at a comparable average image count of approximately 2.8. Figure 9 further shows that incorporating Memory reduces the average number of iterations from 2.98 to 2.80 and shifts the distribution toward earlier termination. Together with the performance gains in the memory ablation, these results indicate that retaining useful evidence and distilled experiences supports both generation quality and earlier stopping. At the system level, GEMS combines higher final scores with fewer image generation calls, providing a favorable trade-off between performance and generation efficiency.

B.2 Results by Benchmark Category

In this section, we provide a fine-grained performance breakdown for GenEval Ghosh et al. (2023), GenEval2 Kamath et al. (2025), DPG-Bench Hu et al. (2024), OneIG Chang et al. (2025), WISE Niu et al. (2025), SpatialGenEval Wang et al. (2026), and CREA Venkatesh et al. (2025), as shown in Table 4 through Table 11 . We report the scores for each individual subset adhering to the official scripts. These detailed results offer a more comprehensive comparison of how different inference-time scaling strategies impact specific generation capabilities.

Table 4: Detailed Performance Breakdown on GenEval. We report the fine-grained scores across six dimensions: Single Object, Two Object, Counting, Colors, Position, and Attribute Binding, with the overall performance shown in the last column.

Base Model	Method	Single Obj	Two Obj	Counting	Colors	Position	Attr Binding	Overall
Closed-Source Models								
Nano Banana	–	–	–	–	–	–	–	0.75
GPT-Image 1	–	0.99	0.92	0.85	0.92	0.75	0.61	0.84
Seedream 4 Seedream et al. (2025)	–	–	–	–	–	–	–	0.84
Open-Source Models								
Bagel Deng et al. (2025)	–	–	–	–	–	–	–	0.82
Z-Image Cai et al. (2025)	–	1.00	0.94	0.78	0.93	0.62	0.77	0.84
Qwen-Image Wu et al. (2025)	–	0.99	0.92	0.89	0.88	0.76	0.77	0.87
Inference-Time Scaling								
Z-Image-Turbo	Original	1.00	0.92	0.73	0.87	0.49	0.61	0.77
	Rewrite	1.00	0.89	0.69	0.89	0.60	0.56	0.77
	Promptist Hao et al. (2023)	1.00	0.91	0.70	0.86	0.51	0.59	0.76
	Search Ma et al. (2025)	1.00	0.92	0.85	0.86	0.60	0.65	0.81
	Maestro Wan et al. (2025)	1.00	0.95	0.80	0.89	0.65	0.62	0.82
	CRAFT Kovalev et al. (2025)	0.99	0.91	0.76	0.88	0.65	0.63	0.80
	GEMS (Ours)	0.99	0.92	0.91	0.88	0.83	0.65	0.86
Qwen-Image-2512	Original	0.95	0.87	0.58	0.77	0.36	0.47	0.66
	Rewrite	0.98	0.85	0.60	0.70	0.56	0.41	0.68
	Promptist Hao et al. (2023)	0.91	0.83	0.54	0.74	0.42	0.54	0.66
	Search Ma et al. (2025)	1.00	0.94	0.61	0.81	0.63	0.47	0.74
	Maestro Wan et al. (2025)	1.00	0.93	0.70	0.73	0.62	0.58	0.76
	CRAFT Kovalev et al. (2025)	1.00	0.90	0.74	0.82	0.70	0.58	0.79
	GEMS (Ours)	0.98	0.94	0.73	0.84	0.72	0.54	0.79

•

Table 5: Detailed Performance Breakdown on GenEval2. We report the fine-grained scores across five dimensions: Object, Attribute, Count, Position, and Verb.

Base Model	Method	Object	Attribute	Count	Position	Verb
Closed-Source Models						
Nano Banana	–	99.0	91.4	70.1	70.2	86.7
Nano Banana 2	–	97.0	95.1	76.4	83.7	56.1
Open-Source Models						
Bagel Deng et al. (2025)	–	92.9	75.9	55.6	50.6	57.8
Qwen-Image Wu et al. (2025)	–	99.1	85.6	70.3	60.2	71.1
Inference-Time Scaling						
Z-Image-Turbo	Original	97.8	75.4	62.8	62.2	18.6
	Rewrite	97.2	86.8	64.9	73.2	27.6
	Promptist Hao et al. (2023)	98.4	78.2	63.1	65.6	23.1
	Search Ma et al. (2025)	98.4	78.1	73.8	67.1	22.9
	Maestro Wan et al. (2025)	96.6	87.1	67.2	75.8	36.0
	CRAFT Kovalev et al. (2025)	98.1	93.1	78.0	86.7	45.6
	GEMS (Ours)	97.9	92.4	79.0	85.4	41.1
Qwen-Image-2512	Original	97.9	65.2	66.9	58.1	22.1
	Rewrite	97.3	81.1	64.0	73.8	30.3
	Promptist Hao et al. (2023)	97.3	64.4	63.0	59.4	28.6
	Search Ma et al. (2025)	98.3	71.1	73.9	66.1	28.2
	Maestro Wan et al. (2025)	97.3	84.9	68.0	80.9	31.0
	CRAFT Kovalev et al. (2025)	97.9	91.3	79.0	89.9	53.7
	GEMS (Ours)	96.9	91.4	81.3	91.4	45.5

Table 6: Detailed Performance Breakdown on DPG-Bench. We report the fine-grained scores across five dimensions: Global, Entity, Attribute, Relation, and Other, with the overall performance shown in the last column.

Base Model	Method	Global	Entity	Attribute	Relation	Other	Overall
Closed-Source Models							
Nano Banana	–	–	–	–	–	–	85.23
GPT-Image 1	–	88.89	88.94	89.84	92.63	90.96	85.15
Seedream 4 Seedream et al. (2025)	–	–	–	–	–	–	88.25
Open-Source Models							
Bagel Deng et al. (2025)	–	–	–	–	–	–	85.10
Z-Image Cai et al. (2025)	–	93.39	91.22	93.16	92.22	91.52	88.14
Qwen-Image Wu et al. (2025)	–	91.32	91.56	92.02	94.31	92.73	88.32
Inference-Time Scaling							
Z-Image-Turbo	Original	90.56	90.40	90.70	91.39	90.58	85.08
	Rewrite	85.23	88.43	90.23	90.18	92.01	84.48
	Promptist Hao et al. (2023)	70.78	72.87	72.02	73.12	73.56	65.69
	Search Ma et al. (2025)	91.34	89.60	90.62	92.51	91.06	85.47
	Maestro Wan et al. (2025)	87.13	91.24	90.66	91.15	91.25	85.29
	CRAFT Kovalev et al. (2025)	93.15	91.41	89.16	91.80	90.38	85.29
	GEMS (Ours)	90.82	89.97	90.48	93.11	90.61	86.01
Qwen-Image-2512	Original	88.36	90.70	90.33	91.57	89.88	84.69
	Rewrite	91.66	89.19	88.42	92.07	88.99	83.49
	Promptist Hao et al. (2023)	72.92	71.97	71.26	71.87	73.38	64.02
	Search Ma et al. (2025)	92.72	90.31	89.34	93.74	89.81	86.17
	Maestro Wan et al. (2025)	92.11	88.79	89.24	89.35	91.62	84.07
	CRAFT Kovalev et al. (2025)	91.34	92.07	88.56	92.74	89.74	85.87
	GEMS (Ours)	87.39	91.81	90.00	93.45	92.37	85.59

Table 7: Detailed Performance Breakdown on OneIG-EN. We report the fine-grained scores across five dimensions: Alignment, Text, Reasoning, Style, and Diversity, with the overall performance shown in the last column.

Base Model	Method	Alignment	Text	Reasoning	Style	Diversity	Overall
Closed-Source Models							
Nano Banana	–	0.878	0.894	0.346	0.450	0.182	0.550
GPT-Image 1	–	0.851	0.857	0.345	0.462	0.151	0.533
Seedream 4 Seedream et al. (2025)	–	0.894	0.981	0.352	0.458	0.197	0.576
Open-Source Models							
Bagel Deng et al. (2025)	–	0.769	0.244	0.173	0.367	0.251	0.361
Z-Image Cai et al. (2025)	–	0.881	0.987	0.280	0.387	0.194	0.546
Qwen-Image Wu et al. (2025)	–	0.882	0.891	0.306	0.418	0.197	0.539
Inference-Time Scaling							
Z-Image-Turbo	Original	0.836	0.987	0.302	0.370	0.137	0.526
	Rewrite	0.835	0.839	0.277	0.433	0.215	0.520
	Promptist Hao et al. (2023)	0.667	0.266	0.264	0.308	0.161	0.333
	Search Ma et al. (2025)	0.845	0.990	0.306	0.376	0.134	0.530
	Maestro Wan et al. (2025)	0.843	0.964	0.294	0.427	0.212	0.548
	CRAFT Kovalev et al. (2025)	0.863	0.980	0.309	0.460	0.300	0.582
	GEMS (Ours)	0.869	0.983	0.312	0.451	0.232	0.569
Qwen-Image-2512	Original	0.830	0.910	0.235	0.233	0.225	0.487
	Rewrite	0.828	0.752	0.249	0.265	0.231	0.465
	Promptist Hao et al. (2023)	0.650	0.166	0.208	0.173	0.242	0.288
	Search Ma et al. (2025)	0.848	0.944	0.248	0.254	0.218	0.502
	Maestro Wan et al. (2025)	0.836	0.921	0.264	0.283	0.166	0.494
	CRAFT Kovalev et al. (2025)	0.858	0.944	0.289	0.331	0.241	0.533
	GEMS (Ours)	0.859	0.972	0.282	0.333	0.262	0.542

Table 8: Detailed Performance Breakdown on OneIG-ZH. We report the fine-grained scores across five dimensions: Alignment, Text, Reasoning, Style, and Diversity, with the overall performance shown in the last column.

Base Model	Method	Alignment	Text	Reasoning	Style	Diversity	Overall
Closed-Source Models							
Nano Banana	–	0.825	0.276	0.298	0.427	0.198	0.405
GPT-Image 1	–	0.812	0.650	0.300	0.449	0.159	0.474
Seedream 4 Seedream et al. (2025)	–	0.847	0.982	0.286	0.443	0.206	0.553
Open-Source Models							
Bagel Deng et al. (2025)	–	0.672	0.365	0.186	0.357	0.268	0.370
Z-Image Cai et al. (2025)	–	0.793	0.988	0.266	0.386	0.243	0.535
Qwen-Image Wu et al. (2025)	–	0.825	0.963	0.267	0.405	0.279	0.548
Inference-Time Scaling							
Z-Image-Turbo	Original	0.773	0.958	0.275	0.362	0.138	0.501
	Rewrite	0.788	0.645	0.269	0.424	0.217	0.469
	Promptist Hao et al. (2023)	0.377	0.510	0.250	0.292	0.207	0.327
	Search Ma et al. (2025)	0.784	0.964	0.280	0.365	0.135	0.506
	Maestro Wan et al. (2025)	0.792	0.896	0.281	0.405	0.220	0.519
	CRAFT Kovalev et al. (2025)	0.805	0.863	0.288	0.446	0.306	0.542
	GEMS (Ours)	0.800	0.978	0.294	0.448	0.240	0.552
Qwen-Image-2512	Original	0.784	0.966	0.243	0.226	0.227	0.489
	Rewrite	0.775	0.658	0.250	0.263	0.231	0.435
	Promptist Hao et al. (2023)	0.405	0.432	0.216	0.189	0.293	0.307
	Search Ma et al. (2025)	0.796	0.976	0.251	0.244	0.217	0.497
	Maestro Wan et al. (2025)	0.785	0.904	0.264	0.258	0.235	0.489
	CRAFT Kovalev et al. (2025)	0.801	0.875	0.274	0.326	0.313	0.518
	GEMS (Ours)	0.810	0.979	0.279	0.325	0.267	0.532

Table 9: Detailed Performance Breakdown on WISE. We report the fine-grained scores across six dimensions: Cultural, Time, Space, Biology, Physics, and Chemistry, with the overall performance shown in the last column.

Base Model	Method	Cultural	Time	Space	Biology	Physics	Chemistry	Overall
Closed-Source Models								
GPT-Image 1	–	0.81	0.71	0.89	0.83	0.79	0.74	0.80
Seedream 4 Seedream et al. (2025)	–	0.78	0.73	0.85	0.79	0.84	0.67	0.78
Open-Source Models								
Bagel Deng et al. (2025)	–	0.76	0.69	0.75	0.65	0.75	0.58	0.70
Qwen-Image Wu et al. (2025)	–	0.62	0.63	0.77	0.57	0.75	0.40	0.62
Inference-Time Scaling								
Z-Image-Turbo	Original	0.57	0.58	0.71	0.50	0.60	0.39	0.57
	Rewrite	0.88	0.78	0.89	0.80	0.81	0.81	0.84
	Promptist Hao et al. (2023)	0.54	0.53	0.68	0.45	0.58	0.39	0.54
	Search Ma et al. (2025)	0.60	0.59	0.78	0.56	0.66	0.44	0.61
	Maestro Wan et al. (2025)	0.88	0.78	0.89	0.81	0.85	0.83	0.85
	CRAFT Kovalev et al. (2025)	0.83	0.72	0.77	0.76	0.76	0.72	0.78
	GEMS (Ours)	0.81	0.76	0.86	0.82	0.81	0.79	0.81
Qwen-Image-2512	Original	0.60	0.61	0.68	0.51	0.66	0.37	0.59
	Rewrite	0.86	0.76	0.86	0.78	0.80	0.71	0.81
	Promptist Hao et al. (2023)	0.56	0.59	0.69	0.52	0.66	0.37	0.57
	Search Ma et al. (2025)	0.67	0.64	0.81	0.64	0.73	0.49	0.67
	Maestro Wan et al. (2025)	0.85	0.79	0.87	0.85	0.86	0.77	0.84
	CRAFT Kovalev et al. (2025)	0.84	0.73	0.81	0.75	0.76	0.76	0.79
	GEMS (Ours)	0.81	0.77	0.86	0.79	0.78	0.77	0.80

Table 10: Detailed Performance Breakdown on SpatialGenEval. We report the fine-grained scores across ten dimensions: Object, Attribute, Position, Orientation, Layout, Comparison, Proximity, Occlusion, Motion, and Causal, with the overall performance shown in the last column.

Base Model	Method	Object	Attribute	Position	Orientation	Layout	Comparison	Proximity	Occlusion	Motion	Causal	Overall
Closed-Source Models												
Nano Banana	–	58.5	75.3	55.5	58.9	70.9	31.8	68.7	33.5	81.4	82.2	61.7
GPT-Image 1	–	56.3	74.1	53.3	58.9	70.4	31.4	66.8	30.2	80.9	82.2	60.5
Seedream 4 Seedream et al. (2025)	–	59.9	80.2	57.2	58.9	70.1	32.1	68.3	33.8	83.0	83.8	62.7
Open-Source Models												
Bagel Deng et al. (2025)	–	55.3	73.7	51.2	54.0	62.9	28.6	64.1	29.0	74.4	76.7	57.0
Qwen-Image Wu et al. (2025)	–	61.0	77.2	55.6	56.7	69.7	28.6	67.7	30.8	78.1	80.2	60.6
Inference-Time Scaling												
Z-Image-Turbo	Original	56.5	74.7	53.2	56.0	67.5	29.9	66.3	29.6	75.8	77.1	58.7
	Rewrite	56.6	74.2	53.0	57.9	67.8	28.9	67.0	30.6	79.7	79.8	59.6
	Promptist Hao et al. (2023)	37.7	65.9	47.6	52.6	56.3	22.4	52.9	24.3	61.2	62.1	48.3
	Search Ma et al. (2025)	56.7	75.8	55.5	57.0	67.9	28.9	66.1	29.8	76.3	79.3	59.3
	Maestro Wan et al. (2025)	59.0	74.7	53.9	58.9	68.3	30.5	67.2	30.8	79.4	81.1	60.3
	CRAFT Kovalev et al. (2025)	56.3	75.9	54.0	57.1	69.7	31.5	67.3	30.8	82.0	81.2	60.6
	GEMS (Ours)	58.9	75.9	55.4	57.4	71.1	32.0	69.0	31.1	81.1	82.1	61.4
Qwen-Image-2512	Original	56.8	76.2	56.8	58.5	67.6	28.5	68.0	31.9	77.2	78.6	60.0
	Rewrite	57.6	75.6	54.3	58.1	68.0	28.1	69.4	32.2	79.8	80.3	60.3
	Promptist Hao et al. (2023)	39.3	65.1	47.6	50.8	55.4	21.9	55.8	25.0	62.9	62.9	48.7
	Search Ma et al. (2025)	56.3	76.5	55.5	60.2	68.6	30.1	69.6	30.6	79.3	79.6	60.6
	Maestro Wan et al. (2025)	56.9	75.5	57.0	59.3	68.7	30.1	69.4	32.6	79.8	81.1	61.0
	CRAFT Kovalev et al. (2025)	59.3	75.8	54.4	60.5	68.5	31.6	68.0	32.6	82.9	82.7	61.6
	GEMS (Ours)	61.3	76.3	57.7	58.9	70.7	31.1	68.8	32.8	81.5	82.0	62.1

Table 11: Detailed Performance Breakdown on CREA. We report the fine-grained scores across six dimensions: Originality, Expressiveness, Aesthetic, Technical, Unexpected, and Interpretability, with the overall performance shown in the last column.

Base Model	Method	Originality	Expressiveness	Aesthetic	Technical	Unexpected	Interpretability	Overall
Inference-Time Scaling								
Z-Image-Turbo	Original	1.62	1.57	2.92	3.19	1.37	1.17	11.84
	Rewrite	2.39	2.63	3.72	3.80	2.22	1.99	16.75
	Promptist Hao et al. (2023)	1.85	2.02	3.36	3.76	1.57	1.45	14.00
	Search Ma et al. (2025)	1.87	1.88	3.07	3.33	1.56	1.28	12.98
	Maestro Wan et al. (2025)	2.32	2.46	3.66	3.76	1.92	1.68	15.81
	CRAFT Kovalev et al. (2025)	1.98	2.01	3.20	3.44	1.57	1.43	13.63
	GEMS (Ours)	3.36	3.48	4.36	4.45	3.57	3.33	22.55
Qwen-Image-2512	Original	2.46	2.81	3.67	3.89	2.36	2.28	17.45
	Rewrite	2.63	2.83	3.96	4.07	2.38	2.25	18.12
	Promptist Hao et al. (2023)	2.54	2.78	3.95	4.41	2.32	2.35	18.35
	Search Ma et al. (2025)	2.69	2.95	3.86	4.07	2.58	2.56	18.71
	Maestro Wan et al. (2025)	2.59	2.76	3.93	4.05	2.34	2.12	17.78
	CRAFT Kovalev et al. (2025)	2.30	2.42	3.57	3.74	1.88	1.70	15.61
	GEMS (Ours)	3.52	3.87	4.59	4.65	3.69	3.68	24.01

C Additional Experiments

C.1 Combining Multiple Skills

To examine whether skills can be used jointly within a single generation task, we extend LongText-Bench by adding a creative-image requirement to each prompt. The resulting tasks combine text rendering with creative drawing. We compare GEMS with at most two triggered skills against GEMS with at most one triggered skill. Claude Opus 4.7 serves as the judge, comparing the generated outputs in terms of rendered text accuracy and overall visual composition. Comparisons judged as ties are excluded when calculating the win and loss percentages below.

Table 12: Pairwise comparison on the extended LongText-Bench prompts, excluding ties. Wins and losses are reported from the perspective of the setting with at most two skills, relative to the setting with at most one skill. Percentages are normalized over comparisons with a decisive judgment.

Comparison	**Win (%)**	**Loss (%)**
At most 2 skills vs. at most 1 skill	63.6	36.4

As shown in Table 12, the two-skill setting wins 63.6% of comparisons with a decisive judgment. This supports the benefit of combining text rendering and creative drawing guidance on these prompts, although it does not establish that arbitrary skill combinations are free of conflicts.

C.2 Alternative MLLM Backend

We evaluate GEMS with Qwen3.5-27B in place of Kimi K2.5 as the MLLM backend on GenEval2, using both Z-Image-Turbo and Qwen-Image-2512 as image generators. Table 13 reports the results under the same evaluation setting. With Qwen3.5-27B, GEMS scores 1.6 and 0.7 points below the corresponding Kimi K2.5 configurations, while remaining substantially above the original generators. These results support the use of an alternative MLLM backend for the two generators on GenEval2.

Table 13: GenEval2 results with different MLLM backends. The columns identify the image generators.

Method	**Z-Image-Turbo**	**Qwen-Image-2512**
Original	31.0	29.0
GEMS (Kimi K2.5)	63.5	70.4
GEMS (Qwen3.5-27B)	61.9	69.7

C.3 Inference Cost

We compare GEMS, CRAFT, and Maestro on GenEval2 using Kimi K2.5 and Qwen-Image-2512. Table 14 reports the average numbers of MLLM calls, input and output tokens, and image generation calls per prompt, together with estimated API costs in USD. The estimates use the official API prices at the time of the experiment. The final row reports the total wall-clock time for the full GenEval2 evaluation set, rather than latency per prompt.

GEMS has the lowest estimated total cost among the three methods in this setting, with fewer image generation calls and output tokens. It uses more MLLM calls than Maestro, illustrating why call counts alone do not capture the total cost. Image generation accounts for the largest cost component for each method. The timing results describe the measured evaluation setup and depend on service throughput and execution conditions.

D Qualitative Results

In addition to Figure 7, we further present comparisons of the results generated by GEMS and the baseline with same prompt, as shown in Figure 16 and Figure 17

Table 14: Inference cost on GenEval2 with Kimi K2.5 and Qwen-Image-2512. All rows except the last are averages per prompt. Costs are in USD; total evaluation time is in minutes.

Metric	**GEMS**	**CRAFT**	**Maestro**
MLLM calls	20.40	27.51	16.37
Input tokens	29938	25246	14010
Output tokens	10056	14765	29296
Image generation calls	2.67	3.77	4.92
Input cost	0.018	0.014	0.008
Output cost	0.031	0.046	0.090
Image generation cost	0.098	0.139	0.181
Total cost	0.147	0.199	0.279
Total evaluation time	232	283	271



Figure 16: Baseline vs GEMS for prompt: *a view of the Earth from the moon.*

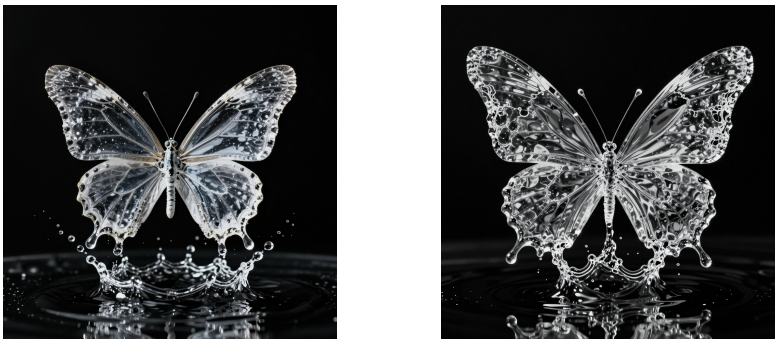


Figure 17: Baseline vs GEMS for prompt: *A high-speed photography shot of clear water being splashed upwards from a black surface. In mid-air, the splashing water droplets perfectly form the shape of a detailed, symmetrical butterfly. The wings are made entirely of liquid water ripples. The background is completely pitch black. The water butterfly is physically connected to the splash below.*

E LIMITATIONS AND FUTURE WORK

First, despite utilizing the lightweight and distilled Z-Image-Turbo, the iterative nature of Agent Loop still results in noticeable inference latency. Future work will focus on optimizing the workflow design to minimize computational overhead and improve overall efficiency.

Second, the current system relies on predefined workflows to coordinate agent collaboration and module interactions. We plan to investigate higher levels of agent harness in the future, such as providing tool interfaces that allow models to autonomously manage memory and skills.

Third, while the current system is primarily designed for image generation, GEMS has the potential to be extended to other multimodal tasks. Future research will explore its application in more complex domains such as video generation.

Finally, current Z-Image-Turbo and Qwen-Image-2512 do not support image editing. Future research could leverage more versatile models to evolve GEMS into a comprehensive Agent-Native system that integrates reasoning, generation, and editing in a unified intelligence loop.