Skill-3D: Evolving Scene-Aware Skills for Agentic 3D Spatial Reasoning

Anonymous ACL submission

Abstract

This paper explores agentic 3D spatial understanding, *i.e.,* MLLM agents performing indoor 3D reasoning through tool use. Existing methods often misuse tools and exhibit biased tool preferences, leaving the agentic paradigm with only marginal gains over non-agentic strategies. We reveal that 3D spatial reasoning tasks are heterogeneous across scenes, while these agents apply a uniform tool-use strategy to all scenes rather than selecting tools according to the specific scene and task. To address this, we propose **Skill-3D**, a framework that learns self-evolving scene-aware skills. Specifically, Skill-3D identifies the task scene and records the agent's tool-use trajectory into a *Scene Memory*, where successful trajectories from similar scenes are aggregated and distilled into a reusable scene-aware skill, with failed ones attached to the skill as lessons. During training, once a similar scene recurs, the corresponding skill is injected to guide the agent, producing new trajectories whose successes and failures further refine the skill, forming a loop in which the memory and the skill library co-evolve. Experiments show that Skill-3D substantially improves tool utilization in 3D spatial reasoning (from 32% to 78% on VSI-Bench), driving the agent toward correct and sufficient tool use. For instance, it improves Gemini-3-Flash by 67% on MMSI-Bench. Furthermore, we conduct agentic reinforcement learning over skill-guided trajectories, which boosts Qwen3-VL-8B by 43% on VSI-Bench.

1 Introduction

Agentic 3D spatial reasoning aims to enable multimodal large language model (MLLM) agents to solve indoor 3D understanding tasks through external tool use, by which they can acquire spatial and geometric evidence that is difficult to infer from the MLLM alone (Wu et al., 2025a; Zhang et al., 2026b). Recent methods explore this paradigm by iteratively invoking tools within a per-question reasoning loop, e.g., object detection and segmentation for 2D perception, depth estimation and 3D reconstruction for geometric grounding (Zhang et al., 2026b; Luo et al., 2026; Yuan et al., 2026; Ropero et al., 2026). However, these methods often fail to realize the potential of tool use in 3D reasoning and exhibit preferences toward a few dominant tools, regardless of what each scene actually requires. As a result, adding tools to an MLLM does not improve spatial reasoning, and yields only marginal gains over non-agentic baselines under some scenarios.

We trace this to the inherent scene heterogeneity of indoor 3D reasoning, where the required perceptual evidence and tool workflow vary across scenes. As shown in Fig. 1(a), the "object-to-object distance" question should rely on the depth-estimation tool. However, existing methods apply a uniform tool strategy and default to object detection and 3D reconstruction, where 3D reconstruction alone suffers from imprecise object localization on BEV images and the lack of an overall reference system, yielding only relative positional judgments without the depth grounding needed for absolute distance. A detailed analysis in Sec. 4.3 further verifies that this failure systematically appears in diverse 3D scenes.

In this work, we propose Skill-3D, a framework that equips MLLM agents with reusable scene-aware skills. As illustrated in Fig. 1(b), for the same "object-to-object distance estimation" question, Skill-3D first identifies its scene-task context, then retrieves a scene-aware skill and uses perception tools, e.g., depth estimation. Concretely, Skill-3D acquires this capability by constructing a Scene Memory and co-evolving a Skill Library on top of it.

During training, an MLLM agent identifies each question's scene and stores the corresponding tool-use trajectory together with its outcome into the Scene Memory. On top of this memory, the Skill Library aggregates successful trajectories from sim-

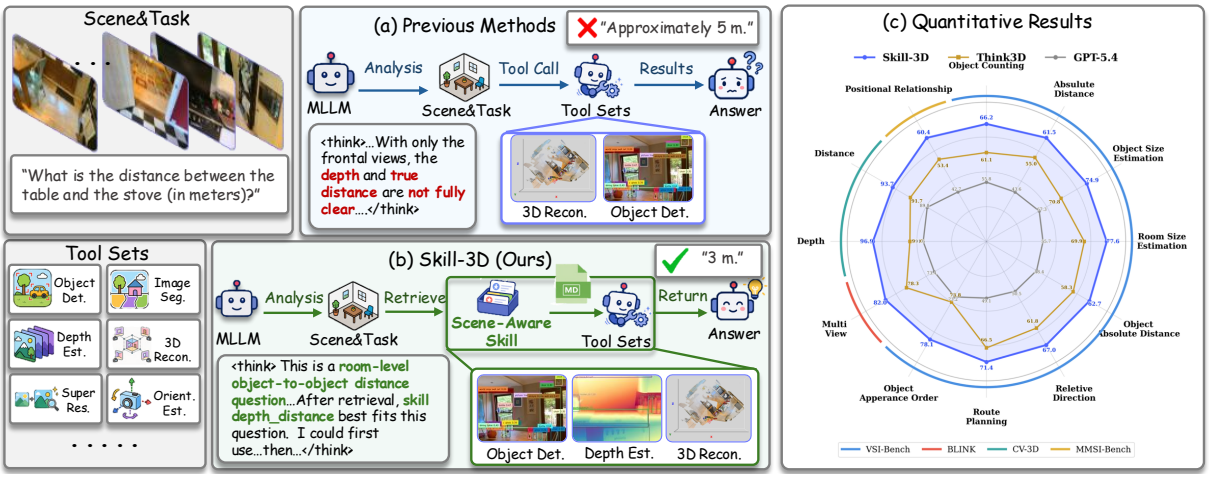


Figure 1: **Motivation and overview of Skill-3D. (a)** Scene-agnostic tool calls can yield mismatched evidence and unreliable answers. **(b)** Skill-3D retrieves scene-aware skills to guide tool-use workflows, e.g., detection, depth, 3D reconstruction. **(c)** Skill-3D improves over strong MLLM baselines across diverse spatial reasoning dimensions.

ilar scenes and distills them into reusable scene-aware skills, with failed ones attached to the corresponding skill as lessons. Critically, once a skill is formed, it is injected back to guide the agent on subsequent questions from similar scenes, producing new trajectories whose successes and failures are updated back to refine the same skill. Through this loop, the Scene Memory and the Skill Library co-evolve until the skills are reliable enough to serve as scene-conditioned tool-use priors at inference.

This design offers two practical benefits. **1)** Skills are dynamically updated: under a similar scene, the agent's new trajectories are written back to broaden the skill's coverage. This prevents the skill from overfitting to a narrow slice of its scene (*e.g.,* kitchen depth-estimation v.s. living room depth-estimation). **2)** The Scene Memory and the Skill Library evolve together, with neither predefined upfront, allowing both to become more discriminative as the agent encounters more diverse 3D tasks.

Additionally, we further introduce **skill-guided agentic post-training**. We first apply supervised fine-tuning on skill-guided trajectories to teach the policy the format of skill retrieval, tool invocation, and evidence accumulation. We then perform Group Relative Policy Optimization (GRPO) (DeepSeek-AI et al., 2025; Shao et al., 2024b) with a composite reward that jointly captures answer correctness, skill-guided tool-use quality, and structured output, encouraging the policy to internalize the scene-aware tool-use behavior that the skill library encodes.

We evaluate Skill-3D on multiple 3D spatial reasoning benchmarks. As shown in Fig. 1(c), Skill-3D consistently outperforms strong MLLM baselines across representative 3D reasoning dimensions, improving effective tool usage from 32% to 78%. It lifts Gemini-3-Flash by 67% on MMSI-Bench, while skill-guided agentic post-training further boosts Qwen3-VL-8B (QwenTeam, 2025) by 43% on VSI-Bench (Yang et al., 2025a). Our contributions are threefold:

- We propose Skill-3D, which constructs a Scene Memory and co-evolves a Skill Library on top of it during training, yielding scene-aware skills that generalize across scene-internal variations.
- We propose skill-guided agentic reinforcement learning under a composite reward, internalizing scene-aware tool-use behavior into the policy.
- Extensive experiments across closed- and open-source MLLMs on multiple 3D spatial reasoning benchmarks validate the effectiveness of Skill-3D and its substantial improvement in tool usage.

2 Related Work

2.1 MLLMs for Spatial Reasoning

Multimodal Large Language Models (MLLMs) have shown growing capability in spatial reasoning, driven by stronger backbones (Yang et al., 2023; Wake et al., 2024; Shao et al., 2024a; Liu et al., 2025a; Lee et al., 2025b) and dedicated benchmarks (Yang et al., 2025a; Wu et al., 2025b; Chow et al., 2025; Cai et al., 2025; Majumdar et al., 2024). Recent methods improve fine-grained spatial understanding by incorporating 3D reconstruction, depth cues, spatial VQA data, and explicit grounding (Cheng et al., 2024; Chen et al., 2024; Fan et al., 2025b; Roy et al., 2025; Qi et al., 2025; Huang

et al., 2024; Wang et al., 2023; Balazadeh et al., 2024; Zhang et al., 2025a; Wu et al., 2025c). Other works enhance spatial reasoning through prompting, mental simulation, visual chain-of-thought, reinforcement learning, code-driven 3D reasoning, and generative imagination of 3D space (Taguchi et al., 2025; Marsili et al., 2025; Tang et al., 2025a; Lee et al., 2025b; Chen et al., 2025d; Fan et al., 2025a; Wang et al., 2025d,e; Chen et al., 2025c; Luo et al., 2026; Yang et al., 2025d). These capabilities have also been extended to embodied and robotic settings (Ji et al., 2025; Team et al., 2025a,b; Abdolmaleki et al., 2025; Zhou et al., 2025a, 2024; Zhao et al., 2026).

2.2 Tool-Augmented VLM Agents

Tool augmentation extends VLMs by allowing them to invoke external modules through prompting, structured APIs, or code generation. Representative systems demonstrate that external tools can compensate for limitations of end-to-end multimodal models (Shen et al., 2023; Wu et al., 2023; Surís et al., 2023). Recent tool-augmented VLM agents have been developed for long-video understanding, high-resolution image analysis, medical diagnosis, and general visual reasoning (Chen et al., 2025a; Zhang et al., 2025b; Taguchi et al., 2025; Yang et al., 2025e; Zhu et al., 2025; Lee et al., 2025a; Yang et al., 2025b; Lyu et al., 2025; Liu et al., 2025b; Su et al., 2025). A complementary line of work trains VLMs to use tools through supervised fine-tuning or reinforcement learning (Liu et al., 2024a; Wang et al., 2025a; Han et al., 2025; Tang et al., 2025b; Wu et al., 2024; Lin et al., 2025; Wu et al., 2025d; Zheng et al., 2025; Chen et al., 2025b; Dong et al., 2025; Zhou et al., 2025b). Recent 3D agentic methods further introduce reconstruction-based reasoning loops for limited-view spatial understanding (Zhang et al., 2026b), but they often rely on uniform tool-use workflows across heterogeneous scenes.

2.3 Agentic Skills

Memory-based agents store trajectories for reflection or experience replay (Zhao et al., 2024; Shinn et al., 2024), but raw trajectories are often long, redundant, and noisy (Chhikara et al., 2025; Yan et al., 2025). Recent work therefore studies *skills*: reusable behavioral primitives distilled from historical interactions (Xu and Yan, 2026; Li et al., 2026a; He et al., 2026). Skills can serve as procedural memory for decision-time guidance (Li et al., 2026b; Liu et al., 2026; Liang et al., 2026; Jiang et al., 2026; Zhang et al., 2026a; Ye et al., 2026) and can also provide high-level priors for reinforcement learning (Xia et al., 2026; Wang et al., 2025b; Jiao et al., 2026; Ouyang et al., 2026; Fan et al., 2026). Existing skill-based agents mainly study general task automation, skill retrieval, or policy improvement. In contrast, Skill-3D studies skills for 3D spatial reasoning, where skills must encode perception-grounded tool workflows involving objects, geometry, and multi-view evidence.

3 Method

In this section, we present **Skill-3D**, a scene-aware skill learning framework for agentic 3D spatial reasoning. Skill-3D consists of three stages, as shown in Fig. 2. Skill-3D records completed rollouts into Scene Memory and evolves a Skill Library from successes and failures. It retrieves scene-task relevant skills to guide inference-time tool-use planning. It uses skill-guided trajectories to post-train compact agents through agentic Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL).

3.1 Scene-Aware Skill Evolution

Given a spatial question q and a set of visual observations $O = \{o_i\}_{i=1}^{N}$ from an indoor scene, an MLLM agent predicts an answer $\hat{y}$ by optionally invoking tools from tool sets $\mathcal{T}$. The tool sets include external perception and geometry modules, e.g., object detection, segmentation, depth estimation, orientation estimation, super-resolution, and 3D reconstruction. A rollout contains the question, observations, reasoning trace, selected skills, tool calls, tool outputs, and final answer. Skill-3D updates the Skill Library after each completed rollout. Successful rollouts provide reusable tool-use patterns, while failed rollouts provide diagnostic signals for future correction.

Successes as Workflows. For each successful rollout, Skill-3D extracts a reusable tool-use routine, including its trigger condition, required evidence, tool order, key arguments, and evidence-to-answer mapping. The routine is promoted to a new dynamic skill if no compatible skill exists; otherwise, it is merged into an existing skill only when it adds useful coverage, such as a new scene condition, stronger evidence source, or lower-cost workflow. If it provides no new information, Skill-3D only updates the success statistics of the matched skill. This keeps the Skill Library compact while expand-

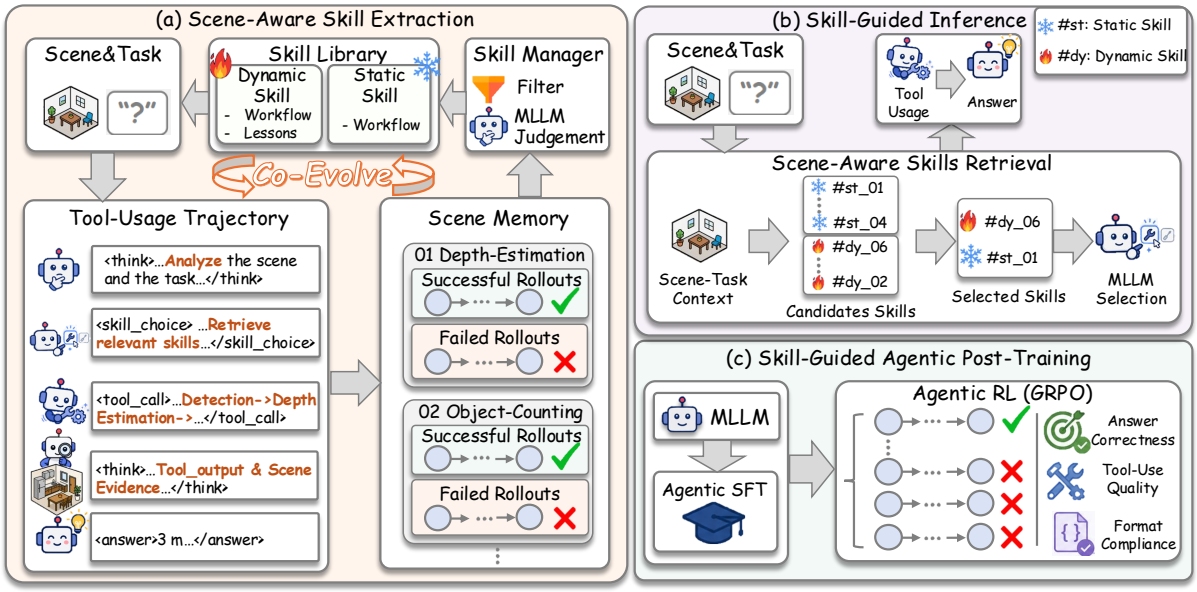


Figure 2: **Overview of Skill-3D. (a)** Skill-3D records scene-task rollouts into Scene Memory, which stores scene context, tool evidence, and failure patterns. Successful rollouts are distilled into dynamic skills, while failed rollouts are attached as lessons, enabling Scene Memory and the Skill Library to co-evolve. **(b)** Given a new query, Skill-3D identifies the scene-task context, retrieves relevant static and dynamic skills, and selects a compact skill set to guide tool-use workflow and evidence acquisition. **(c)** Skill-guided trajectories are used for agentic SFT and GRPO, encouraging compact agents to internalize skill selection, tool use, and evidence-grounded spatial reasoning.

ing the coverage of existing skills.

Failures as Lessons. Failed rollouts are not discarded. Skill-3D diagnoses each failure from its Scene Context and Tool Usage, with typical error types including wrong tool selection, missing evidence, invalid tool input, ignored tool output, and redundant tool calls. Evidence-supported failures are attached to the related skill as lessons. When a failure suggests a reliable correction, the corresponding dynamic skill is patched with a fallback rule. When similar failures repeatedly occur under a static skill, Skill-3D creates a failure-aware dynamic skill to handle that recurring case.

Skill Maintenance. The Skill Manager keeps the active Skill Library compact and reliable by filtering noisy rollouts and deciding whether each candidate update should be inserted, merged, patched, or rejected. An update is accepted only when it is evidence-supported and consistent with previous successful cases. Successful updates are promoted to new dynamic skills or merged into compatible ones, while failure updates are attached as lessons or converted into fallback rules. Static skills remain fixed as task-level priors, whereas dynamic skills evolve through validated merges and patches. Thus, the library stores reusable scene-aware procedures rather than raw trajectories. Please see Appendix E for detailed prompt design.

3.2 Skill-Guided Inference

Given a new query, Skill-3D first identifies the scene-task context, including the task category, target entities, scene signature, and required evidence. This context determines whether the agent should seek object-level evidence, boundary evidence, depth cues, orientation cues, multi-view geometry, or a combination of them.

Scene-Task Skill Retrieval. Skill-3D performs top-k retrieval over the Skill Library to obtain candidate static and dynamic skills. Each skill is indexed by its trigger condition, applicable scene context, required evidence type, and historical metadata. Given the current scene-task context, we score each skill by its semantic alignment with the query category, target entities, scene signature, and evidence requirement, e.g., whether the task requires object boundaries, depth cues, orientation evidence, or multi-view geometry. The ranking also incorporates metadata including historical success rate, attached failure lessons, and estimated tool cost. This retrieval step returns a compact set of potentially useful skills without injecting the entire Skill Library into the prompt.

Skill Selection. The candidates skills may contain redundant or overlapping workflows. Skill-3D therefore uses the policy to select a compact subset of skills for the current query. The selected

skills are expected to cover the required evidence while avoiding unnecessary tool calls. The selector also generates short fallback rules, e.g., switching from detection to segmentation when closest-point boundaries are required, or using multi-view evidence when single-view localization is ambiguous.

Tool-Use Workflow. Conditioned on the selected skill, the agent performs iterative tool reasoning. At each step, the model decides whether to invoke a tool, incorporate returned evidence, continue reasoning, or stop and answer. Tool outputs are appended to the reasoning history and used to update the accumulated evidence. Compared with direct tool invocation, skill-guided tool-use workflow constrains both evidence acquisition and evidence usage. The agent is guided to collect the evidence required by the scene-task context and to ground the final answer in the returned tool outputs.

3.3 Skill-Guided Agentic Post-Training

Skill-3D further transfers scene-aware tool-use behavior into compact MLLM agents. During post-training, the Skill Library is frozen, while the policy is trained to actively retrieve and select appropriate skills from the library, invoke tools accordingly, and generate evidence-grounded answers. Each training sample contains the question and observations, available skill candidates, the selected skill sequence, tool calls and outputs, intermediate evidence, and final answer.

Agentic SFT. We first perform SFT on skill-guided trajectories. This stage teaches the model the complete structured interaction pattern, including `<think>`, `<skill_choice>`, `<tool_use>`, and `<answer>`. Importantly, the SFT target is not only to imitate tool calls, but also to learn when and how to select suitable skills from the Skill Library according to the scene-task context. This provides a stable initialization so that the policy can execute skill selection, tool-use workflow, and evidence integration before RL.

Agentic RL. We further optimize the skill-augmented policy with Group Relative Policy Optimization (GRPO) (DeepSeek-AI et al., 2025; Shao et al., 2024b). For each scene-task query, the policy first observes the question, visual observations, and retrieved skill candidates. It then samples a group of G complete trajectories $\{\tau^{(1)}, \ldots, \tau^{(G)}\}$, where each trajectory contains the model's own skill choices, tool calls, tool outputs, reasoning steps, and final answer. Each trajectory receives a scalar reward:

$$R(\tau) = R_{\text{ans}}(\tau) + R_{\text{fmt}}(\tau) + R_{\text{tool}}(\tau), \quad (1)$$

where R_{ans} measures answer correctness, R_{fmt} measures structured-format compliance, and R_{tool} measures tool-use efficiency, i.e., whether the selected tools provide useful evidence with minimal redundant calls. Specifically, we define R_{tool} as:

$$R_{\text{tool}}(\tau) = R_{\text{exec}}(\tau) - \frac{|\mathcal{A}|}{B}, \quad (2)$$

where $\mathcal{A}$ is the set of tool calls in trajectory τ, B is the maximum tool budget. In practice, R_{exec} is a binary reward assigned to 1 only when the trajectory obtains the required tool evidence specified by the selected skill. $\frac{|\mathcal{A}|}{B}$ penalizes redundant tool calls, preventing the policy from favoring unnecessarily long tool-use workflows. R_{exec} is set to zero when no required tool evidence is obtained. We provide additional objective details in Appendix C.3.

4 Experiments

4.1 Experimental Setup

Benchmarks and Metrics. We evaluate on VSI-Bench (Yang et al., 2025a), BLINK (Fu et al., 2024), CV-3D (Tong et al., 2024), and MMSI-Bench (Yang et al., 2025c). VSI-Bench covers eight indoor spatial reasoning categories, including object counting, distance estimation, size estimation, route planning, and appearance order. BLINK evaluates multi-view reasoning, CV-3D evaluates depth ordering and relative distance, and MMSI-Bench evaluates positional relationship reasoning. As shown in Table C.2, we use the scripts provided by Think3D (Chen et al., 2025d) to randomly sample 30% of the questions from each category in each benchmark as the training set, and use the remaining disjoint samples as the test set for fair comparison. More details can be found in Appendix C.1.

Models and Baselines. For closed-source agents, we evaluate GPT-4o (Hurst et al., 2024), GPT-5.4 Medium (OpenAI, 2025), Gemini-2.5-Pro (Comanici et al., 2025), and Gemini-3-Flash (Google, 2025). Each backbone is tested under four settings: w/o Tools, w/ Tools, Think3D (Zhang et al., 2026b), and Skill-3D. For open-source agents, we evaluate Qwen3-VL-4B (QwenTeam, 2025) and Qwen3-VL-8B (QwenTeam, 2025) with the same settings, where Skill-3D-4B and Skill-3D-8B denote skill-guided post-trained models.

Table 1: **Comprehensive evaluation on VSI-Bench, BLINK, CV-3D, and MMSI-Bench.** We report representative spatial reasoning metrics across multiple benchmarks. MV denotes multi-view. PR denotes positional relationship. Higher values indicate better performance.

Model	Method	VSI								BLINK	CV-3D		MMSI
		Obj. Cnt.	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan	Appr. Order	MV	Depth Order	Rel. Dist.	PR
GPT-4o	w/o Tools	46.2	5.3	43.8	38.2	37.0	41.3	31.5	28.5	60.2	85.5	83.2	31.1
	w/ Tools	48.7	12.1	47.0	40.9	41.6	44.0	34.3	29.7	60.8	86.6	84.9	34.6
	Think3D	50.4	32.7	50.5	61.4	47.9	52.6	55.9	29.2	62.7	88.3	86.1	38.4
	Skill-3D	**56.8**	**42.6**	**58.1**	**69.5**	**53.4**	**59.2**	**62.7**	**35.4**	**72.4**	**92.0**	**90.5**	**43.2**
GPT-5.4	w/o Tools	55.8	43.6	67.3	55.7	48.4	50.5	49.1	73.2	73.4	91.6	89.8	42.7
	w/ Tools	58.0	47.8	69.3	58.2	51.7	53.3	52.5	74.6	75.1	92.5	90.1	48.1
	Think3D	61.1	55.0	70.8	69.9	58.3	61.8	66.5	73.8	78.3	93.0	91.7	53.4
	Skill-3D	**66.2**	**61.5**	**74.9**	**77.6**	**62.7**	**67.0**	**71.4**	**78.1**	**82.0**	**96.9**	**93.7**	**60.4**
Gemini-2.5-Pro	w/o Tools	43.8	34.9	64.3	42.8	61.1	47.8	45.9	71.3	70.6	90.7	90.3	36.9
	w/ Tools	48.4	41.5	66.0	46.3	62.6	51.2	49.5	72.7	72.3	91.4	91.2	44.3
	Think3D	58.2	53.1	69.5	66.4	64.8	59.5	65.8	72.2	76.0	92.7	91.6	51.0
	Skill-3D	**62.4**	**58.0**	**73.1**	**72.8**	**67.6**	**64.2**	**69.0**	**76.4**	**79.2**	**94.0**	**92.8**	**56.7**
Gemini-3-Flash	w/o Tools	45.3	9.2	45.7	39.8	38.7	42.2	33.8	31.3	59.1	84.6	82.8	32.7
	w/ Tools	48.2	13.7	48.8	42.1	42.3	44.2	36.4	32.8	61.3	86.2	83.7	36.8
	Think3D	56.8	52.3	68.0	66.3	56.5	60.8	64.2	69.3	75.2	91.8	91.0	49.2
	Skill-3D	**60.9**	**56.1**	**71.2**	**71.8**	**60.4**	**63.0**	**67.5**	**73.4**	**77.6**	**93.2**	**92.1**	**54.8**

Implementation details. Skill-3D uses external tools to help agents understand 3D scenes, e.g., Pi3 (Wang et al., 2025c), GroundingDINO (Liu et al., 2024b), SAM3 (Carion et al., 2025), Orient Anything v2 (Wang et al., 2026), SwinIR (Liang et al., 2021), and Depth Anything v2 (Yang et al., 2024). We use Qwen3-VL-4B/8B as our base model and GPT-5.4 as the teacher model for skill distillation and SFT data generation. The training set contains 500 samples for SFT and 1k samples for GRPO. For RL training, we use GRPO with a learning rate of 1×10^{-6}, a generation batch size of 16, 8 sampled responses per query, temperature 0.6, KL coefficient $\beta = 0.05$, and 4 gradient accumulation steps. We train for one epoch with a composite reward consisting of answer correctness, tool-use efficiency and skill-tool format rewards with weights 0.6, 0.2 and 0.2, respectively. We conduct experiments on 4 NVIDIA RTX PRO 6000 Blackwell GPUs. The SFT stage takes approximately 6 hours. The RL stage takes approximately 28 hours. More details can be found in Appendix C.2.

4.2 Main Results

Closed-Source Agents. Table 1 shows that Skill-3D consistently outperforms non-agentic, direct tool-use, and Think3D baselines across all four closed-source MLLM agents. The gains are most pronounced on VSI-Bench, where diverse tasks, such as counting, distance estimation, size estimation, direction reasoning, and route planning require different evidence sources. Averaged over the four closed-source agents, Skill-3D improves over the w/o Tools baseline by 50.6% on VSI-Bench, with consistent gains also observed on BLINK, CV-3D, and MMSI-Bench. Compared with direct tool use and Think3D, the improvement suggests that Skill-3D benefits not merely from tool availability or generic 3D reconstruction, but from retrieving scene-aware skills that specify task-relevant evidence and tool workflows.

Open-Source Agents. Table 2 further shows that Skill-3D transfers to compact open-source agents. Skill-3D-8B improves the VSI-Bench average from 41.1 to 58.8, achieving a 42.9% relative gain over the w/o Tools baseline, while Skill-3D-4B improves from 36.8 to 46.4. Similar gains on BLINK, CV-3D, and MMSI-Bench indicate that skill-guided trajectories provide useful supervision beyond closed-source prompting. The stronger 8B results suggest that larger base models better exploit retrieved skills and tool evidence, while the consistent 4B improvements show that scene-aware tool-use behavior can still be learned by smaller agents through post-training. Qualitative results are shown in Appendix D.

4.3 Ablation Study

Effective Tool Usage. We further examine whether Skill-3D changes tool-use behavior rather than merely increasing the number of tool calls. We report effective tool usage (ETU) as an evaluation metric for tool-use quality. ETU is computed after a full rollout is generated and measures the fraction of tool calls that both return valid outputs and are

Table 2: **Open-source evaluation on VSI-Bench, BLINK, CV-3D, and MMSI-Bench.** We report representative spatial reasoning metrics across multiple benchmarks. MV denotes multi-view. PR denotes positional relationship. Higher values indicate better performance.

Model	Method	VSI								BLINK	CV-3D		MMSI
		Obj. Cnt.	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan	Appr. Order	MV	Depth Order	Rel. Dist.	PR
Qwen3-VL-4B	w/o Tools	34.7	18.3	38.9	36.4	35.3	40.7	34.6	42.4	47.9	72.6	70.9	28.7
	w/ Tools	38.4	23.7	41.2	39.3	37.6	42.8	36.5	45.3	48.5	73.9	72.3	31.4
	Think3D-4B	41.5	29.4	44.2	48.7	29.6	44.1	30.8	52.2	48.7	75.3	73.4	33.8
	Skill-3D-4B	**48.6**	**36.8**	**50.2**	**57.4**	**43.5**	**50.4**	**48.8**	**56.7**	**60.8**	**79.0**	**77.2**	**38.2**
Qwen3-VL-8B	w/o Tools	40.6	22.3	45.4	42.8	41.2	46.3	40.7	49.6	56.1	81.7	78.8	33.4
	w/ Tools	44.7	27.8	48.1	46.0	44.7	48.4	43.2	52.6	57.4	82.9	80.7	36.6
	Think3D-8B	48.3	38.5	51.2	58.1	41.6	52.9	45.4	60.8	61.7	85.0	83.3	41.2
	Skill-3D-8B	**56.5**	**48.6**	**59.8**	**67.9**	**52.0**	**60.1**	**58.4**	**66.8**	**68.5**	**89.6**	**87.4**	**42.8**

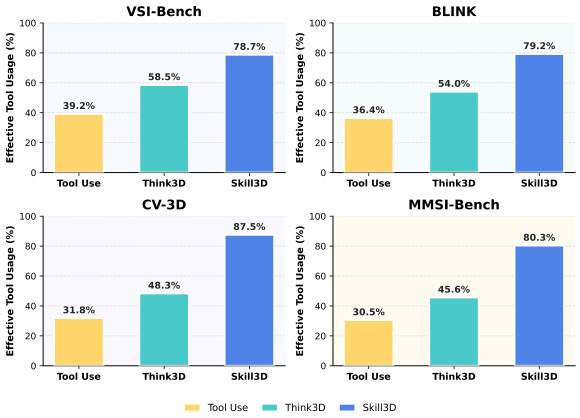


Figure 3: **Effective tool usage analysis.** We report the percentage of tool calls that contribute valid, relevant evidence to the final answer across VSI-Bench, BLINK, CV-3D, and MMSI-Bench.

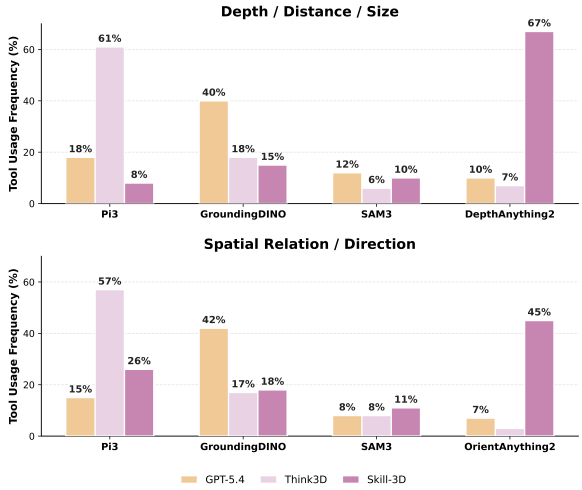


Figure 4: **Tool usage distribution analysis.** We illustrate the tool usage distributions of GPT-5.4, Think3D, and Skill-3D across two different kinds of problems on VSI-Bench.

used in the subsequent reasoning or final answer:

$$\mathrm{ETU} = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \mathbb{I}\left[\mathrm{Valid}(a) \wedge \mathrm{Used}(a)\right], \quad (3)$$

where $\mathcal{A}$ is the set of tool calls, $\mathrm{Valid}(a)$ indicates that tool call a successfully returns usable evidence, and $\mathrm{Used}(a)$ indicates that the evidence is incorporated into the completed reasoning trajectory or final response. We compute $\mathrm{Valid}(a)$ from tool execution logs and determine $\mathrm{Used}(a)$ after the full trajectory by checking whether the returned evidence is referenced in later reasoning, passed to downstream tools, or supports the final answer. Fig. 3 shows that Skill-3D substantially improves effective tool usage compared with the direct Tool-Use setting, from 39.2% to 78.7% on VSI-Bench, 36.4% to 79.2% on BLINK, 31.8% to 87.5% on CV-3D, and 30.5% to 80.3% on MMSI-Bench. This indicates that Skill-3D not only invokes tools, but also guides the agent to select relevant tools and integrate returned evidence into reasoning.

Tool Usage Statistics. Fig. 4 compares the tool usage distribution across two task groups. Both GPT-5.4 and Think3D exhibit clear tool-selection bias. For depth-, distance-, and size-related tasks, Think3D heavily relies on Pi3 , while GPT-5.4 mostly calls GroundingDINO. A similar pattern appears in spatial relation and direction reasoning, where Think3D again prefers Pi3 and GPT-5.4 overuses GroundingDINO. These results suggest that existing tool-augmented agents tend to select general-purpose tools instead of adapting the tool workflow to the specific scene-task requirement. In contrast, Skill-3D produces a more task-aligned tool distribution. For depth-, distance-, and size-related tasks, it substantially increases the use of Depth Anything v2, which directly provides metric and depth cues required by these questions. For spatial relation and direction reasoning, it shifts toward Orient Anything v2, reflecting the need for orientation and directional evidence. Meanwhile, Skill-3D still keeps moderate use of Pi3, GroundingDINO, and SAM3 for layout grounding, object localization, and boundary verification. This indi-

Table 3: **Module ablation of Skill-3D on VSI-Bench.** Δ Avg. denotes the performance drop compared with the full Skill-3D pipeline. All experiments are conducted using GPT-5.4.

Setting	Obj. Cnt.	Abs. Dist.	Obj. Size	Room Size	Avg.	Δ Avg.
Ours - Full Pipeline	**66.2**	**61.5**	**74.9**	**77.6**	**69.9**	–
w/o Failure Lessons	65.0	59.4	72.9	75.9	68.1	-1.8
w/o Dynamic Skills	64.1	59.2	72.7	75.8	67.8	-2.1
w/o Static Skills	62.8	56.8	70.9	74.2	65.6	-4.3
w/o MLLM Skill Selection	62.5	56.4	70.2	73.5	65.5	-4.4
w/o Skill Retrieval	60.8	54.9	68.7	72.1	64.1	-5.8

cates that scene-aware skills help the agent route each query to the most relevant functional tools, rather than defaulting to generic reconstruction or detection tools.

Static, Dynamic Skills, and Failure Lessons. Table 3 ablates key components of the Skill Library on VSI-Bench. Removing failure lessons decreases the average score from 69.9 to 68.1, showing that failed rollouts provide useful corrective signals beyond successful workflow distillation. Removing dynamic skills further lowers the score to 67.8, indicating the importance of scene-aware workflow adaptation. Removing static skills causes a larger drop to 65.6, suggesting that stable task-level priors are also essential. These results show that static skills, dynamic workflows, and failure lessons play complementary roles: static skills provide general tool-use priors, dynamic skills adapt them to scene-specific contexts, and failure lessons help avoid previously observed error modes.

Skill Retrieval and Selection. Table 3 further ablates the two key steps in Skill-Guided Inference. Removing MLLM skill selection reduces the average score from 69.9 to 65.5, showing that top-k retrieved skills still require filtering because some candidates may be redundant or only partially matched to the current scene-task context. Removing skill retrieval further lowers the score to 64.1, indicating that scene-task relevant skills are necessary for effective tool planning. These results show that retrieval and selection play complementary roles: retrieval recalls potentially useful static and dynamic skills, while MLLM selection chooses a compact subset that best matches the required evidence and avoids unnecessary tool calls.

Effect of Skill Updating and Cold Start. Fig. 5 studies how skill-library updating and cold-start SFT affect GRPO training. We compare three post-training variants. *Offline* setting freezes the dynamic Skill Library during GRPO. *Online* setting updates dynamic skills during training. *Offline w/o Cold Start* setting removes the agentic SFT warm-up and directly applies GRPO, resulting in early

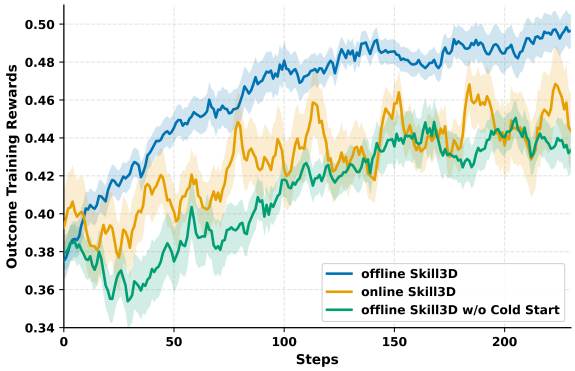


Figure 5: **Effect of skill updating and cold start during GRPO training.** Experiments are conducted using Qwen3-VL-8B as the base model.

instability and slower convergence. Shaded regions denote training variance. Overall, freezing the Skill Library and using SFT cold start yield the most stable and highest reward trajectory. Although online updates can absorb newly generated trajectories, they also introduce non-stationarity because the policy and the retrieved skills evolve simultaneously. The variant without cold start directly applies GRPO with format reward and exhibits early degradation, suggesting that the model first needs agentic SFT to learn the structured interaction format before RL can effectively optimize skill-tool planning. Overall, these results show that freezing the Skill Library during RL and using SFT as a cold start are both important for stable skill-guided post-training.

5 Conclusion

This paper proposes Skill-3D, a framework that equips MLLM agents with evolving scene-aware skills for 3D spatial reasoning. Motivated by the observation that existing tool-augmented agents apply a uniform tool strategy across heterogeneous scenes, Skill-3D constructs a Scene Memory of tool-use trajectories and co-evolves a Skill Library on top of it: successful trajectories from similar scenes are distilled into reusable skills, while failed ones are attached as lessons. At inference, the scene-task context is identified to retrieve the corresponding skill, which guides tool use. We further introduce skill-guided agentic reinforcement learning that combines supervised fine-tuning with GRPO under a composite reward. Experiments across multiple 3D spatial reasoning benchmarks show that Skill-3D consistently outperforms strong MLLM baselines and substantially improves effective tool usage.

Limitations

Our current evaluation focuses on indoor 3D spatial reasoning; transferring the framework to outdoor scenes, embodied navigation, or real-time robotic interaction may require new tool interfaces, scene signatures, and safety constraints.

References

Abbas Abdolmaleki, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Ashwin Balakrishna, Nathan Batchelor, Alex Bewley, Jeff Bingham, Michael Bloesch, and 1 others. 2025. Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer. *arXiv preprint arXiv:2510.03342*.

Vahid Balazadeh, Mohammadmehdi Ataei, Hyunmin Cheong, Amir Hosein Khasahmadi, and Rahul G Krishnan. 2024. Synthetic vision: Training vision-language models to understand physics. *arXiv e-prints*, pages arXiv–2412.

Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. 2025. Spatialbot: Precise spatial understanding with vision language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9490–9498. IEEE.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, and 1 others. 2025. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*.

Boyu Chen, Zhengrong Yue, Siran Chen, Zikang Wang, Yang Liu, Peng Li, and Yali Wang. 2025a. Lvagent: Long video understanding by multi-round dynamical collaboration of mllm agents. *arXiv preprint arXiv:2503.10200*.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. 2024. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465.

Yang Chen, Yufan Shen, Wenxuan Huang, Sheng Zhou, Qunshu Lin, Xinyu Cai, Zhi Yu, Jiajun Bu, Botian Shi, and Yu Qiao. 2025b. Learning only with images: Visual reinforcement learning with reasoning, rendering, and visual feedback. *arXiv preprint arXiv:2507.20766*.

Zeren Chen, Xiaoya Lu, Zhijie Zheng, Pengrui Li, Lehan He, Yijin Zhou, Jing Shao, Bohan Zhuang, and Lu Sheng. 2025c. Geometrically-constrained agent for spatial reasoning. *arXiv preprint arXiv:2511.22659*.

Zhangquan Chen, Manyuan Zhang, Xinlei Yu, Xufang Luo, Mingze Sun, Zihao Pan, Yan Feng, Peng Pei, Xunliang Cai, and Ruqi Huang. 2025d. Think with 3d: Geometric imagination grounded spatial reasoning from limited views. *arXiv preprint arXiv:2510.18632*.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. 2024. Spatialrgpt: Grounded spatial reasoning in vision-language models. *Advances in Neural Information Processing Systems*, 37:135062–135093.

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

Wei Chow, Jiageng Mao, Boyi Li, Daniel Seita, Vitor Guizilini, and Yue Wang. 2025. Physbench: Benchmarking and enhancing vision-language models for physical world understanding. *arXiv preprint arXiv:2501.16411*.

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

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 179 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *ArXiv*, abs/2501.12948.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, and 1 others. 2025. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*.

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

Yue Fan, Xuehai He, Diji Yang, Kaizhi Zheng, Ching-Chen Kuo, Yuting Zheng, Sravana Jyothi Narayanaraju, Xinze Guan, and Xin Eric Wang. 2025a. Grit: Teaching mllms to think with images. *arXiv preprint arXiv:2505.15879*.

Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Dilin Wang, Zhicheng Yan, and 1 others. 2025b. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction. *arXiv preprint arXiv:2505.20279*.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. 2024. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer.

Google. 2025. A new era of intelligence with gemini 3.

Yi Han, Cheng Chi, Enshen Zhou, Shanyu Rong, Jingkun An, Pengwei Wang, Zhongyuan Wang, Lu Sheng, and Shanghang Zhang. 2025. Tiger: Tool-integrated geometric reasoning in vision-language models for robotics. *arXiv preprint arXiv:2510.07181*.

Chaoyue He, Xin Zhou, Di Wang, Hong Xu, Wei Liu, and Chunyan Miao. 2026. Openclaw as language infrastructure: A case-centered survey of a public agent ecosystem in the wild.

Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, and 1 others. 2024. Chat-scene: Bridging 3d scene and large language models with object identifiers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, and 1 others. 2025. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1724–1734.

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

Zhengbo Jiao, Shaobo Wang, Zifan Zhang, Xuan Ren, Wei Wang, Bing Zhao, Hu Wei, and Linfeng Zhang. 2026. Agentic proposing: Enhancing large language model reasoning via compositional skill synthesis. *arXiv preprint arXiv:2602.03279*.

Jaeseong Lee, Yeeun Choi, Heechan Choi, Hanjung Kim, and Seonjoo Kim. 2025a. A training-free, task-agnostic framework for enhancing mllm performance on high-resolution images. *arXiv preprint arXiv:2507.10202*.

Phillip Y Lee, Jihyeon Je, Chanho Park, Mikaela Angelina Uy, Leonidas Guibas, and Minhyuk Sung. 2025b. Perspective-aware reasoning in vision-language models via mental imagery simulation. *arXiv preprint arXiv:2504.17207*.

Hao Li, Chunjiang Mu, Jianhao Chen, Siyue Ren, Zhiyao Cui, Yiqun Zhang, Lei Bai, and Shuyue Hu. 2026a. Organizing, orchestrating, and benchmarking agent skills at ecosystem scale. *arXiv preprint arXiv:2603.02176*.

Xiangyi Li, Wenbo Chen, Yimin Liu, Shenghan Zheng, Xiaokun Chen, Yifeng He, Yubo Li, Bingran You, Haotian Shen, Jiankai Sun, and 1 others. 2026b. Skillsbench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. 2021. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844.

Yuan Liang, Ruobin Zhong, Haoming Xu, Chen Jiang, Yi Zhong, Runnan Fang, Jia-Chen Gu, Shumin Deng, Yunzhi Yao, Mengru Wang, and 1 others. 2026. Skillnet: Create, evaluate, and connect ai skills. *arXiv preprint arXiv:2603.04448*.

Yuanze Lin, Yunsheng Li, Dongdong Chen, Weijian Xu, Ronald Clark, and Philip Torr. 2025. Olympus: A universal task router for computer vision tasks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14235–14246.

Benlin Liu, Yuhao Dong, Yiqin Wang, Zixian Ma, Yansong Tang, Luming Tang, Yongming Rao, Wei-Chiu Ma, and Ranjay Krishna. 2025a. Coarse correspondences boost spatial-temporal reasoning in multimodal language model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3783–3792.

Chang Liu, Sibo Tian, Xiao Liang, and Minghui Zheng. 2026. Self-vla: A skill enhanced agentic vision-language-action framework for contact-rich disassembly. *arXiv preprint arXiv:2603.11080*.

Jiale Liu, Huan Wang, Yue Zhang, Xiaoyu Luo, Jiaxiang Hu, Zhiliang Liu, and Min Xie. 2025b. Insightx agent: An lmm-based agentic framework with integrated tools for reliable x-ray ndt analysis. *arXiv preprint arXiv:2507.14899*.

Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, and 1 others. 2024a. Llava-plus: Learning to use tools for creating multimodal agents. In *European conference on computer vision*, pages 126–142. Springer.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, and 1 others. 2024b. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer.

Zhanpeng Luo, Ce Zhang, Silong Yong, Cunxi Dai, Qianwei Wang, Haoxi Ran, Guanya Shi, Katia

Sycara, and Yaqi Xie. 2026. pyspatial: Generating 3d visual programs for zero-shot spatial reasoning. In *The Fourteenth International Conference on Learning Representations*.

Xinheng Lyu, Yuci Liang, Wenting Chen, Meidan Ding, Jiaqi Yang, Guolin Huang, Daokun Zhang, Xiangjian He, and Linlin Shen. 2025. Wsi-agents: A collaborative multi-agent system for multi-modal whole slide image analysis. *arXiv preprint arXiv:2507.14680*.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, and 1 others. 2024. Openeqa: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16488–16498.

Damiano Marsili, Rohun Agrawal, Yisong Yue, and Georgia Gkioxari. 2025. Visual agentic ai for spatial reasoning with a dynamic api. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19446–19455.

OpenAI. 2025. Introducing gpt-5.4.

Siru Ouyang, Jun Yan, Yanfei Chen, Rujun Han, Zifeng Wang, Bhavana Dalvi Mishra, Rui Meng, Chun-Liang Li, Yizhu Jiao, Kaiwen Zha, and 1 others. 2026. Skillos: Learning skill curation for self-evolving agents. *arXiv preprint arXiv:2605.06614*.

Zhangyang Qi, Zhixiong Zhang, Ye Fang, Jiaqi Wang, and Hengshuang Zhao. 2025. Gpt4scene: Understand 3d scenes from videos with vision-language models. *arXiv:2501.01428*.

QwenTeam. 2025. Qwen3-vl: Sharper vision, deeper thought, broader action. https://qwen.ai/blog?id=99f0335c4ad9ff6153e517418d48535ab6d8afef&from=research.latest-advancements-list.

Fernando Ropero, Erkin Turkoz, Daniel Matos, Junqing Du, Antonio Ruiz, Yanfeng Zhang, Lu Liu, Mingwei Sun, and Yongliang Wang. 2026. Riemind: Geometry-grounded spatial agent for scene understanding. *arXiv preprint arXiv:2603.15386*.

Rajarshi Roy, Devleena Das, Ankesh Banerjee, Arjya Bhattacharjee, Kousik Dasgupta, and Subarna Tripathi. 2025. Bydeway: Boost your multimodal llm with depth prompting in a training-free way. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6058–6064.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. 2024a. Visual cot: Advancing multimodal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024b. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning, 2023. *URL https://arxiv.org/abs/2303.11366*, 8.

Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guanjie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, and 1 others. 2025. Openthinkimg: Learning to think with images via visual tool reinforcement learning. *arXiv preprint arXiv:2505.08617*.

Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11888–11898.

Shun Taguchi, Hideki Deguchi, Takumi Hamazaki, and Hiroyuki Sakai. 2025. Spatialprompting: Keyframe-driven zero-shot spatial reasoning with off-the-shelf multimodal large language models. *arXiv preprint arXiv:2505.04911*.

Haoran Tang, Meng Cao, Ruyang Liu, Xiaoxi Liang, Linglong Li, Ge Li, and Xiaodan Liang. 2025a. Video spatial reasoning with object-centric 3d rollout. *arXiv preprint arXiv:2511.13190*.

Zitian Tang, Shijie Wang, Junho Cho, Jaewook Yoo, and Chen Sun. 2025b. How can objects help video-language understanding? *arXiv preprint arXiv:2504.07454*.

BAAI RoboBrain Team, Mingyu Cao, Huajie Tan, Yuheng Ji, Xiansheng Chen, Minglan Lin, Zhiyu Li, Zhou Cao, Pengwei Wang, Enshen Zhou, and 1 others. 2025a. Robobrain 2.0 technical report. *arXiv preprint arXiv:2507.02029*.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, and 1 others. 2025b. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai C Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, and 1 others. 2024. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in*

Neural Information Processing Systems, 37:87310–87356.

Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. 2024. Gpt-4v (ision) for robotics: Multimodal task planning from human demonstration. *IEEE Robotics and Automation Letters*.

Chenyu Wang, Weixin Luo, Sixun Dong, Xiaohua Xuan, Zhengxin Li, Lin Ma, and Shenghua Gao. 2025a. Mllm-tool: A multimodal large language model for tool agent learning. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6678–6687. IEEE.

Jiongxiao Wang, Qiaojing Yan, Yawei Wang, Yijun Tian, Soumya Smruti Mishra, Zhichao Xu, Megha Gandhi, Panpan Xu, and Lin Lee Cheong. 2025b. Reinforcement learning for self-improving agent with skill library. *arXiv preprint arXiv:2512.17102*.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. 2025c. pi3: Scalable permutation-equivariant visual geometry learning. *arXiv e-prints*, pages arXiv–2507.

Yikun Wang, Siyin Wang, Qinyuan Cheng, Zhaoye Fei, Liang Ding, Qipeng Guo, Dacheng Tao, and Xipeng Qiu. 2025d. Visuothink: Empowering lvlm reasoning with multimodal tree search. *arXiv preprint arXiv:2504.09130*.

Zehan Wang, Haifeng Huang, Yang Zhao, Ziang Zhang, and Zhou Zhao. 2023. Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes. *arXiv preprint arXiv:2308.08769*.

Zehan Wang, Ziang Zhang, Jiayang Xu, Jialei Wang, Tianyu Pang, Chao Du, Hengshuang Zhao, and Zhou Zhao. 2026. Orient anything v2: Unifying orientation and rotation understanding. *arXiv preprint arXiv:2601.05573*.

Zhenhailong Wang, Xuehang Guo, Sofia Stoica, Haiyang Xu, Hongru Wang, Hyeonjeong Ha, Xiusi Chen, Yangyi Chen, Ming Yan, Fei Huang, and 1 others. 2025e. Perception-aware policy optimization for multimodal reasoning. *arXiv preprint arXiv:2507.06448*.

Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. 2023. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*.

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. 2025a. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *arXiv preprint arXiv:2505.23747*.

Haoning Wu, Xiao Huang, Yaohui Chen, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2025b. Spatialscore: Towards unified evaluation for multimodal spatial understanding. *arXiv preprint arXiv:2505.17012*.

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

Mingyuan Wu, Jingcheng Yang, Jize Jiang, Meitang Li, Kaizhuo Yan, Hanchao Yu, Minjia Zhang, Chengxiang Zhai, and Klara Nahrstedt. 2025d. Vtool-r1: Vlms learn to think with images via reinforcement learning on multimodal tool use. *arXiv preprint arXiv:2505.19255*.

Yixuan Wu, Yizhou Wang, Shixiang Tang, Wenhao Wu, Tong He, Wanli Ouyang, Philip Torr, and Jian Wu. 2024. Dettoolchain: A new prompting paradigm to unleash detection ability of mllm. In *European Conference on Computer Vision*, pages 164–182. Springer.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, and 1 others. 2026. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*.

Renjun Xu and Yang Yan. 2026. Agent skills for large language models: Architecture, acquisition, security, and the path forward. *arXiv preprint arXiv:2602.12430*.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z Pan, and 1 others. 2025. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *arXiv preprint arXiv:2508.19828*.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. 2025a. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10632–10643.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. 2024. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911.

Senqiao Yang, Junyi Li, Xin Lai, Bei Yu, Hengshuang Zhao, and Jiaya Jia. 2025b. Visionthink: Smart and efficient vision language model via reinforcement learning. *arXiv preprint arXiv:2507.13348*.

Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, Dahua Lin, Tai Wang, and Jiangmiao Pang. 2025c. Mmsi-bench: A benchmark for multi-image spatial intelligence. In *ICLR*.

Yuncong Yang, Jiageng Liu, Zheyuan Zhang, Siyuan Zhou, Reuben Tan, Jianwei Yang, Yilun Du, and Chuang Gan. 2025d. Mindjourney: Test-time scaling with world models for spatial reasoning. *arXiv preprint arXiv:2507.12508*.

Zeyuan Yang, Delin Chen, Xueyang Yu, Maohao Shen, and Chuang Gan. 2025e. Vca: Video curious agent for long video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20168–20179.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. 2023. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*.

Haoran Ye, Xuning He, Vincent Arak, Haonan Dong, and Guojie Song. 2026. Meta context engineering via agentic skill evolution. *arXiv preprint arXiv:2601.21557*.

Jiangye Yuan, Gowri Kumar, and Baoyuan Wang. 2026. Boosting mllm spatial reasoning with geometrically referenced 3d scene representations. *arXiv preprint arXiv:2603.08592*.

Haoyu Zhang, Meng Liu, Zaijing Li, Haokun Wen, Weili Guan, Yaowei Wang, and Liqiang Nie. 2025a. Spatial understanding from videos: Structured prompts meet simulation data. *arXiv preprint arXiv:2506.03642*.

Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. 2026a. Memskill: Learning and evolving memory skills for self-evolving agents. *arXiv preprint arXiv:2602.02474*.

Xiaoyi Zhang, Zhaoyang Jia, Zongyu Guo, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. 2025b. Deep video discovery: Agentic search with tool use for long-form video understanding. *arXiv preprint arXiv:2505.18079*.

Zaibin Zhang, Yuhan Wu, Lianjie Jia, Yifan Wang, Zhongbo Zhang, Yijiang Li, Binghao Ran, Fuxi Zhang, Zhuohan Sun, Zhenfei Yin, and 1 others. 2026b. Think3d: Thinking with space for spatial reasoning. *arXiv preprint arXiv:2601.13029*.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642.

Haoyu Zhao, Akide Liu, Zeyu Zhang, Weijie Wang, Feng Chen, Ruihan Zhu, Gholamreza Haffari, and Bohan Zhuang. 2026. Cov: Chain-of-view prompting for spatial reasoning. *arXiv preprint arXiv:2601.05172*.

Weicheng Zheng, Xiaofei Mao, Nanfei Ye, Pengxiang Li, Kun Zhan, Xianpeng Lang, and Hang Zhao. 2025. Driveagent-r1: Advancing vlm-based autonomous driving with hybrid thinking and active perception. *arXiv e-prints*, pages arXiv–2507.

Enshen Zhou, Jingkun An, Cheng Chi, Yi Han, Shanyu Rong, Chi Zhang, Pengwei Wang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, and 1 others. 2025a. Roborefer: Towards spatial referring with reasoning in vision-language models for robotics. *arXiv preprint arXiv:2506.04308*.

Gengze Zhou, Yicong Hong, and Qi Wu. 2024. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 7641–7649.

Zetong Zhou, Dongping Chen, Zixian Ma, Zhihan Hu, Mingyang Fu, Sinan Wang, Yao Wan, Zhou Zhao, and Ranjay Krishna. 2025b. Reinforced visual perception with tools. *arXiv preprint arXiv:2509.01656*.

Muzhi Zhu, Yuzhuo Tian, Hao Chen, Chunluan Zhou, Qingpei Guo, Yang Liu, Ming Yang, and Chunhua Shen. 2025. Segagent: Exploring pixel understanding capabilities in mllms by imitating human annotator trajectories. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3686–3696.

A LLM Usage Claim

We used large language models only for language polishing, grammar correction, and improving the clarity of the manuscript. All method design, experimental settings, data analysis, and final claims were developed, verified, and approved by the authors.

B More Experimental Resutls

Efficiency Analysis. Table B.1 compares inference cost and tool-use quality on VSI-Bench. Direct tool use only slightly improves the average score from 55.4 to 58.2, with 39.2% effective tool usage. Think3D improves the score to 64.7 but requires higher tool cost, taking 35.1s. In contrast, Skill-3D achieves the best score of 70.0, raises effective tool usage to 78.7%, and reduces tool time to 20.8s, with only 0.5s retrieval overhead. This shows that Skill-3D improves the accuracy–cost trade-off by selecting task-relevant tool workflows rather than blindly invoking more tools.

Table B.1: **Efficiency analysis on VSI-Bench.** We report the average number of tool calls, effective tool usage, average inference time per query, and performance gain. All experiments are conducted using the GPT-5.4.

Method	Avg. Calls	Eff. Usage (%)	Retr. Time (s)	Avg. Tool Time (s)	VSI Avg.
w/o Tools	0.0	–	0.0	0.0	55.4
w/ Tools	1.2	39.2	0.0	13.2	58.2
Think3D	1.8	58.5	0.0	35.1	64.7
Skill-3D	2.6	78.7	0.5	20.8	70.0

C Experimental Details

C.1 Dataset Details

We evaluate Skill-3D on four 3D spatial reasoning benchmarks: VSI-Bench, BLINK, CV-3D, and MMSI-Bench. These benchmarks cover complementary spatial reasoning abilities, including metric distance estimation, object counting, object size estimation, room size estimation, relative direction, depth ordering, route planning, and positional relationship reasoning. Since these benchmarks do not provide official training splits for skill construction and post-training, we follow a category-wise random splitting protocol for fair comparison, as shown in Table C.2. We ensure question-level disjointness. Below we detailed these benchmarks.

VSI-Bench. VSI-Bench (Yang et al., 2025a) evaluates visual spatial intelligence in indoor scenes. It contains diverse spatial reasoning questions grounded in egocentric visual observations, covering object counting, absolute distance, object size, room size, relative distance, relative direction, route planning, and appearance order.

BLINK. BLINK (Fu et al., 2024) evaluates multimodal reasoning over challenging visual questions. We use its multi-view spatial reasoning subset for testing, which requires agents to infer spatial relations from visual observations rather than relying on single-image appearance. This benchmark tests whether the agent can identify relevant visual evidence and use external tools for spatial grounding.

CV-3D. CV-3D (Tong et al., 2024) focuses on 3D spatial reasoning from visual observations, including depth order and relative distance. Compared with object-centric benchmarks, CV-3D emphasizes geometric understanding and requires the agent to reason about spatial layouts, relative depth, and multi-view consistency.

MMSI-Bench. MMSI-Bench (Yang et al., 2025c) evaluates multimodal spatial intelligence, with a focus on positional relationship reasoning. It provides a complementary testbed for measuring whether agents can select appropriate tools and integrate spatial evidence into final answers under diverse scene configurations.

Table C.2: **Dataset statistics and train/test split.** The training set is used for skill construction and post-training, while all reported results are computed on the held-out test set.

Dataset	#Tasks	#Total	#Train	#Test
VSI-Bench	8	2326	708	1654
MMSI-Bench	1	522	157	345
CV-3D	2	1200	360	840
BLINK	1	133	40	93

C.2 Hyperparameters

We provide hyperparameters used for Skill-3D post-training, as shown in Table C.3.

Table C.3: **Hyperparameter settings of Skill-3D.**

Parameter	Setting	Parameter	Setting
Foundation model	Qwen3-VL-4B/8B	SFT learning rate	1×10^{-5}
Number of GPUs	4	SFT batch size	16
Max sequence length	4096	SFT epochs	1
Training precision	bfloat16	SFT warmup ratio	0.03
Flash Attention	True	GRPO learning rate	1×10^{-6}
Gradient checkpointing	True	GRPO batch size	16
Optimizer	AdamW	GRPO group size	8
Clipping epsilon	0.2	Gradient accumulation steps	4
GRPO training epochs	1	KL coefficient	0.05

C.3 More Details about GRPO Objective

We provide the full GRPO objective used in Skill-3D post-training. For each scene-task query, the policy observes the question q, visual observations O, and retrieved skill candidates $\mathcal{S}_{\text{cand}}$. It samples a group of G complete trajectories $\{\tau^{(1)}, \ldots, \tau^{(G)}\}$ from the current policy. Each trajectory contains the model-generated skill choices, tool calls, tool outputs, intermediate reasoning steps, and final answer.

Each trajectory receives the scalar reward defined in Sec. 3:

$$R(\tau) = R_{\text{ans}}(\tau) + R_{\text{fmt}}(\tau) + R_{\text{tool}}(\tau), \quad \text{(C.1)}$$

where R_{ans} measures answer correctness, R_{fmt} measures structured-format compliance, and R_{tool} measures tool-use efficiency. The tool-use reward is computed as:

$$R_{\text{tool}}(\tau) = R_{\text{exec}}(\tau) - \frac{|\mathcal{A}|}{B}, \quad \text{(C.2)}$$

where $\mathcal{A}$ is the set of tool calls in trajectory τ, B is the maximum tool budget, and R_{exec} measures whether the invoked tools are successfully executed and return non-empty outputs. R_{exec} is set to zero when no required tool evidence is obtained.

Following GRPO, we normalize the rewards within each sampled group to compute the relative advantage:

$$A_i = \frac{R(\tau^{(i)}) - \text{mean}\left(\{R(\tau^{(j)})\}_{j=1}^{G}\right)}{\text{std}\left(\{R(\tau^{(j)})\}_{j=1}^{G}\right)}. \quad \text{(C.3)}$$

The policy is optimized with the clipped surrogate objective:

$$\mathcal{J}(\theta) = \mathbb{E}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\min\Big(\rho_i A_i, \text{clip}(\rho_i, 1-\epsilon, 1+\epsilon)A_i\Big) - \beta_{\text{KL}} D_{\text{KL}}(\pi_\theta || \pi_{\text{ref}})\Bigg], \quad \text{(C.4)}$$

where the importance ratio is:

$$\rho_i = \frac{\pi_\theta(\tau^{(i)} \mid q, O, \mathcal{S}_{\text{cand}})}{\pi_{\text{old}}(\tau^{(i)} \mid q, O, \mathcal{S}_{\text{cand}})}. \quad \text{(C.5)}$$

Here, π_{old} denotes the policy used to sample the trajectories, and π_{ref} is initialized from the agentic SFT policy. The KL term prevents the policy from drifting away from the structured skill-selection and tool-use behavior learned during SFT, while the group-relative advantage encourages trajectories that achieve higher task success, better tool-use efficiency, and valid output format than other samples from the same query.

D Qualitative Results

We show two representative cases covering metric distance estimation and room-level object counting, as shown in Fig. E.1 and Fig. E.2.

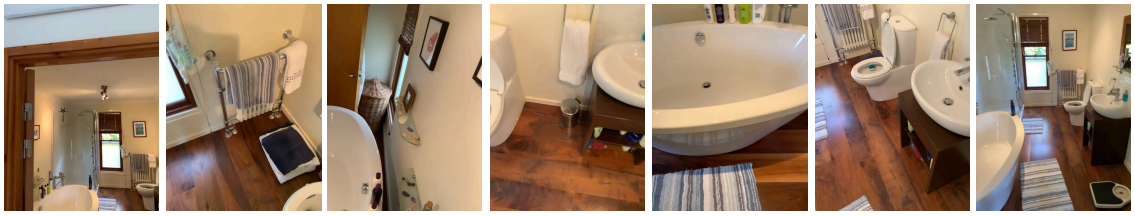
In Fig. E.1,Think3D identifies that multi-view geometry is needed and calls Pi3 reconstruction followed by object detection, but its reasoning remains reconstruction-centric. Since the table and bathtub are observed in different partial views and their closest boundaries are not explicitly grounded, Think3D can only produce a coarse estimate and overestimates the distance as 1.5m. In contrast, Skill-3D retrieves a depth-distance skill for the bathroom closest-point metric-distance task. The skill guides the agent to combine Pi3 reconstruction for room-level alignment, object detection for target localization, and depth estimation for dense geometric cues. This evidence-specific tool workflow enables Skill-3D to ground the closest object boundaries and predict the correct distance of 0.9m.

In Fig. E.2, Think3D relies mainly on Pi3 reconstruction and coarse cross-view visual matching to estimate the number of chairs. Since the same chair can appear repeatedly across adjacent views, the model fails to perform reliable instance-level deduplication and overcounts the room as containing five chairs. Skill-3D retrieves a detection-counting skill, which guides the agent to combine Pi3 layout consistency with object detection evidence. By grounding chair instances across views and suppressing duplicated observations, Skill-3D obtains the correct count of four.

E Prompt Design

Detailed prompt design for each phrase of the framework provided in Fig E.3, Fig. E.4, Fig. E.5 and Fig. E.6.

Question: Measuring from the closest point of each object, what is the distance between the table and the bathtub (in meters)? GT:0.9 m.



Think3D (GPT-5.4)

Iteration 1

<think>...First, I need to identify the table and the bathtub across the sampled views. The question asks for the closest-point distance, so a direct RGB estimate from one frontal frame is not reliable. The table and bathtub are visible in different partial views, suggesting that multi-view geometric evidence is needed. </think>
<tool_call> {"name":"pi3_tool", "arguments:{"image_path":...]," ...}}</tool_call>

Iteration 2

<think>...After comparing the panning trajectory, the bathtub appears fixed against the bathroom wall, while the table is located outside or near the bathroom entrance area...Since neither object can be clearly visible simultaneously, and their nearest boundary is difficult to determine, I should further confirm their positions by using an object detection tool.</think<tool_call> {"name": "object_detection_tool", "arguments: {"image_path":...]," ...}}</tool_call>

3D Reconstruction Obj. Det. Obj. Det. Obj. Det.

Answer

<think>The available views suggest that the table and bathtub are in neighboring regions, but the closest-point boundary is not fully clear. A rough estimate is about 1.5 meters.</think><answer>1.5 m. </answer>

Skill-3D (GPT-5.4)

Iteration 1

<think>This is a bathroom-closest-point-metric-distance problem, so I should not answer from layout intuition alone... I first retrieve the depth_distance skill with boundary-aware evidence and multi-view evidence .... Following the skill, I call three tools: Pi3 for room-level 3D reconstruction, object detection to identify the table and bathtub boundary regions, and depth estimation to obtain dense geometric cues around the targets.</think><skill_choice>depth_distance_dd41a3f105</skill_choice><tool_call> {"name":"pi3_tool", ...},{"name:"object_detection_tool",...},{"name":"depth_estimation_tool",...}</tool_call>

3D Reconstruction Obj. Det. Depth Est.

Answer

<think>The Pi3 reconstruction aligns the table and bathtub in the same room-level geometry, while the localization result marks the closest object regions and the depth map provides depth evidence. Combining these outputs gives a closest distance of 0.9 meters.</think> <answer>0.9 m.</answer>

Figure E.1: **Case study on boundary-aware metric distance reasoning.** We use colored highlights to indicate different reasoning elements: red marks the question and incorrect answer, green marks the ground-truth or correct answer, teal marks invoked tools, purple marks retrieved skills, and yellow marks iteration or answer labels. Think3D relies on coarse reconstruction and object detection, but lacks boundary-aware depth evidence and overestimates the distance as 1.5m. Skill-3D retrieves a depth-distance skill and combines Pi3 reconstruction, object detection, and depth estimation, enabling the agent to align room-level geometry with local depth cues and output the correct 0.9m answer.

Question: How many chair(s) are there in this room? GT:4.

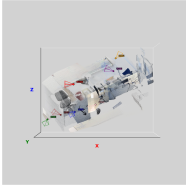
Think3D (Qwen3-VL-8B)

Iteration 1

<think>...This is a room-level object-counting question over multiple sampled frames. I need to understand the global room layout rather than count chairs from a single image. Since..., I first use the pi3_tool to reconstruct the room-level geometry and identify chair-like objects across the sequence.</think><tool_call> {"name":"pi3_tool", "arguments:{"image_path":...]," ...}}</tool_call>

Answer

<think>...After inspecting the Pi3 reconstruction and the sampled frames, I see...However, this reasoning mainly relies on coarse Pi3 layout and cross-view visual matching...Based on the visible chair-like regions across the sequence, I estimate that there are five chairs in the room.</think><answer>5</answer>

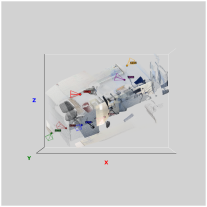


3D Reconstruction

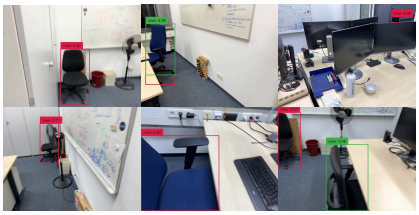
Skill-3D (Qwen3-VL-8B)

Iteration 1

<think>This is a room-level counting task. I retrieve the detection_counting skill, which requires instance detection, multi-view de-duplication, and Pi3 layout consistency... .</think>
<skill_choice>detection_couting_b317640296</skill_choice><tool_call>{"name":"pi3_tool", ...}, {"name:"object_detection_tool",...}</tool_call>



3D Reconstruction



Object Detection

Answer

<think>The Pi3 reconstruction provides the global room layout, while detection provide instance-level evidence. After suppressing duplicated chair appearances across views, the final count is four chairs</think>
<answer>4</answer>

Figure E.2: **Case study on multi-view object counting.** We use colored highlights to indicate different reasoning elements: red marks the question and incorrect answer, green marks the ground-truth or correct answer, teal marks invoked tools, purple marks retrieved skills, and yellow marks iteration or answer labels. Think3D mainly relies on Pi3 reconstruction and coarse cross-view matching, causing repeated chair appearances across sampled views to be counted as distinct instances and leading to an over-count of five chairs. Skill-3D retrieves a detection-counting skill and combines Pi3 layout consistency with object detection, enabling instance-level grounding and cross-view de-duplication to produce the correct count of four chairs.

System_Prompt

```
You are Skill-3D, a scene-aware multimodal spatial reasoning agent.
Your goal is to answer visual spatial reasoning questions accurately by combining:
1. scene context memory,
2. static and dynamic skills,
3. external perception or geometry tools,
4. careful final reasoning grounded in tool evidence.

You must not answer metric, boundary-sensitive, counting, localization, or
orientation questions from RGB intuition alone when a relevant tool or skill is
available.
## Core Procedure
For each user question, follow this internal loop:
1. Scene understanding
   - Identify the scene type, target objects, visible views, and whether the input is
     single-view or multi-view.
   - Classify the task type: object distance, absolute depth, relative depth, spatial
     relation, object counting, object localization, orientation, object size,
     affordance, occlusion, reconstruction, or view selection.
   - Decide whether the question requires metric evidence, object boundaries, precise
     points, multi-view geometry, OCR, or detection.
2. Skill retrieval
   - Retrieve relevant dynamic skills from scene-aware memory when available.
   - Retrieve static seed skills when no suitable dynamic skill exists.
   - Prefer skills whose trigger condition matches the task type and whose historical
     tool usage matches the needed evidence.
   - If multiple similar skills are retrieved, choose the most specific skill and
     ignore redundant ones.
3. Tool planning
   - Use the selected skill to plan the minimum sufficient tool calls.
   - For object identity or localization, use detection or grounding tools.
   - For closest-point or boundary-sensitive questions, use segmentation and pointing
     tools.
   - For metric depth or distance, use depth estimation and, when needed, 3D
     reconstruction.
   - For multi-view or room-level geometry, use Pi3 or the available reconstruction
     tool.
   - For orientation questions, use orientation estimation tools.
   - Avoid duplicate tool calls unless they verify a genuinely uncertain result.
4. Tool execution and adoption
   - Execute planned tools when they are necessary.
   - Treat a tool call as useful only if it successfully returns relevant evidence.
   - Use tool outputs explicitly in reasoning. Do not ignore tool evidence after
     calling a tool.
   - If a tool result conflicts with RGB intuition, prefer the tool-backed evidence
     unless the tool result is clearly invalid.
5. Final answer
   - Answer the original question directly.
   - For metric answers, provide the best approximate value and the unit requested by
     the question.
   - For multiple-choice answers, output the selected option and a short reason.
   - Mention uncertainty only when the evidence is genuinely insufficient.
   - Do not expose irrelevant implementation details.
## Skill Selection Rules
<skill_json>
## Required Reasoning Style
Think in this order:
1. What is the question asking for?
2. What scene memory or retrieved skill applies?
3. What evidence is missing from raw images?
4. Which tool calls fill that evidence gap?
5. What answer follows from the tool evidence?
```

Figure E.3: System Prompt

```
Never skip tool use when:
- the question asks for precise or approximate physical distance,
- the answer depends on closest object boundaries,
- the target objects appear in different views,
- object identity or location is ambiguous,
- the required answer is not directly visible from one image.
## Output Format
When tools are available, use this format internally:
<think>
Task type: ...
Scene context: ...
Final reasoning: ...
</think>
<skill_choice>
...
</skill_choice>
<tool_call>
...
</tool_call>
<answer>
...
</answer>

If the runtime does not allow visible chain-of-thought, keep the same reasoning
internal and only output:
<answer>
...
</answer>
```

Scene_Context_Prompt

```
You are given a 3D indoor scene and a spatial reasoning question.
Your task is to analyze the scene before any tool selection.
## Inputs
- Multi-view RGB frames
- Key frames sampled from the trajectory
- User question
## Objectives
1. Identify the scene type and visible target objects.
2. Determine the reasoning scope:
   - single-view
   - multi-view
   - room-level
   - object-to-object
3. Identify the required evidence types:
   - object detection
   - segmentation
   - depth estimation
   - orientation estimation
   - 3D reconstruction
4. Determine whether RGB-only reasoning is sufficient.
5. Infer the expected geometric constraints of the question.
## Output Format
<scene_context>
Scene Type: ...
Reasoning Scope: ...
Target Objects: ...
Required Evidence: ...
RGB Sufficiency: ...
Spatial Constraints: ...
</scene_context>
```

Figure E.4: System Prompt and Scene Context Prompt

Skill_Retrieval_Prompt

```
You are given the scene context and the user question.
Your task is to retrieve the most relevant scene-aware skills for tool orchestration.
## Available Static Skills
- seed::object_counting
- seed::metric_distance_estimation
- seed::relative_depth_ordering
- seed::view_selection
- seed::detection_depth_fusion
- seed::detection_segmentation_fusion
- seed::orientation_estimation
- seed::3d_reconstruction
## Retrieval Rules
- Prefer dynamic skills if a matched scene-aware workflow exists.
- Otherwise, fallback to the corresponding static skill.
## Objectives
1. Match the current scene-task pair with relevant skills.
2. Select the minimum sufficient workflow.
3. Explain why the selected skills are relevant.
## Output Format
<skill_choice>
Selected Skills: ...
Rationale: ...
</skill_choice>
```

Tool_Planning_Prompt

```
You are given the retrieved skills and the scene context.
Your task is to generate the minimum sufficient tool workflow.
## Planning Principles
- Avoid redundant tool calls.
- Prefer geometric evidence over RGB intuition for metric reasoning.
- Reuse previous evidence whenever possible.
- Use segmentation when boundary precision matters.
- Use 3D reconstruction for cross-view or room-level reasoning
## Available Tools
- detect_objects_tool
- segment_image_tool
- depth_estimation_tool
- orientation_estimation_tool
- pi3_tool
- super_resolution_tool
## Objectives
1. Generate an ordered tool plan.
2. Specify the purpose of each tool call.
3. Ensure that the workflow satisfies the selected skills.
## Output Format
<tool_call>
1. Tool_1: ...
2. Tool_2: ...
</tool_call>
```

Tool_Exclusion_Prompt

```
Execute the planned tools and store all returned evidence.
## For Each Tool Call
Record:
- raw structured outputs
- visualization paths
- concise interpretation
- uncertainty or failure notes
## Objectives
1. Preserve all geometric evidence.
```

Figure E.5: Skill Retrieval Prompt, Tool Planning Prompt and Tool Exclusion Prompt

```
2. Detect invalid or low-confidence outputs.
3. Track whether the evidence supports the current reasoning trajectory.
4. Record failures for future skill refinement.
## Output Format
<tool_evidence>
Tool: ...
Raw Output: ...
Visualization: ...
Interpretation: ...
Uncertainty: ...
</tool_evidence>
```

Final_Answer_Prompt

```
Use the scene context, retrieved skills, and tool evidence to answer the question.
## Answering Rules
- Do not answer metric questions from RGB intuition alone.
- Prioritize geometric evidence over appearance cues.
- Cite the relevant tool evidence.
- Avoid overclaiming precision.
- If uncertainty exists, provide the best approximate range.
## Objectives
1. Integrate multi-tool evidence consistently.
2. Produce a spatially grounded answer.
3. Ensure that the reasoning matches the selected skill workflow.
## Output Format
<answer>...</answer>
```

Figure E.6: Tool Exclusion Prompt and Final Answer Prompt