How Foundational Skills Influence VLM-based Embodied Agents: A Native Perspective

Bo Peng[*1], Bu Pi[*4], Xinrun Xu[2,3], Keyu Pan[4], Miao Chen[4],

Yang Du[4], Lin Li[4], Jun Song[4], Tong Xu[◇1], Bo Zheng[4]

[1]University of Science and Technology of China
[2]Institute of Software, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences
[4]Independent Reasearcher

Abstract

Recent advances in vision–language models (VLMs) have shed light on human-level embodied intelligence. However, existing benchmark for VLM-driven embodied agent still rely on pre-defined high-level command or discretised action spaces—"non-native" settings that diverge markedly from the real world. Moreover, current benchmarks focus exclusively on high-level tasks, while lacking collaborative evaluation and analysis on both low- and high-level. To bridge these gaps, we present **NativeEmbodied**, a challenging benchmark for VLM-driven embodied agents that adopts a unified, native low-level action space. Built upon diverse simulated scenes, NativeEmbodied first designs three representative high-level tasks in complex scenarios to evaluate overall performance. For more detailed and comprehensive performance analysis, we further decouple the entangled skills behind complex tasks and construct four types of low-level tasks, each corresponding to a key fundamental embodied skill. This joint evaluation across task and skill granularities enables a fine-grained assessment of embodied agent. Comprehensive experiments on the best VLMs reveal pronounced deficiencies in certain fundamental embodied skills. Further analysis shows that these low-level bottlenecks severely constrain performance on high-level tasks. Our NativeEmbodied not only pinpoints the key challenges faced by current VLM-driven embodied agents, but also provides valuable insight for future development.

1 Introduction

Recent advances in Vision-Language Models (VLMs) have catalyzed significant progress in embodied intelligence Wang et al. (2024), bringing us closer to intelligent agents that can operate in the simulator or physical world Cheang et al. (2025); Wang et al. (2025); Open-X et al. (2025); Brohan et al. (2023). These VLM-based embodied agents, capable of perceiving the environment through visual inputs, and perform complex task following natural language instructions Chen et al. (2025); Tan et al. (2025); Cao et al. (2025); Long et al. (2025); Yue et al. (2025).

However, a fundamental challenge persists: How can we assess whether these models truly possess the capability to function in the real world, and which fundamental skills bottleneck their performance? This question becomes particularly important as current evaluation benchmarks for embodied agent exhibit several limitations: 1) **Non-Native Action Space**: Recent benchmarks Cheng et al. (2025); Yang et al. (2025) attempt to deploy VLM-based agents in embodied simulators and evaluate their performance through interactive tasks. They typically abstract low-level actions into high-level commands or functions that the agent can invoke directly (e.g., "look at the apple", "teleport to the

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

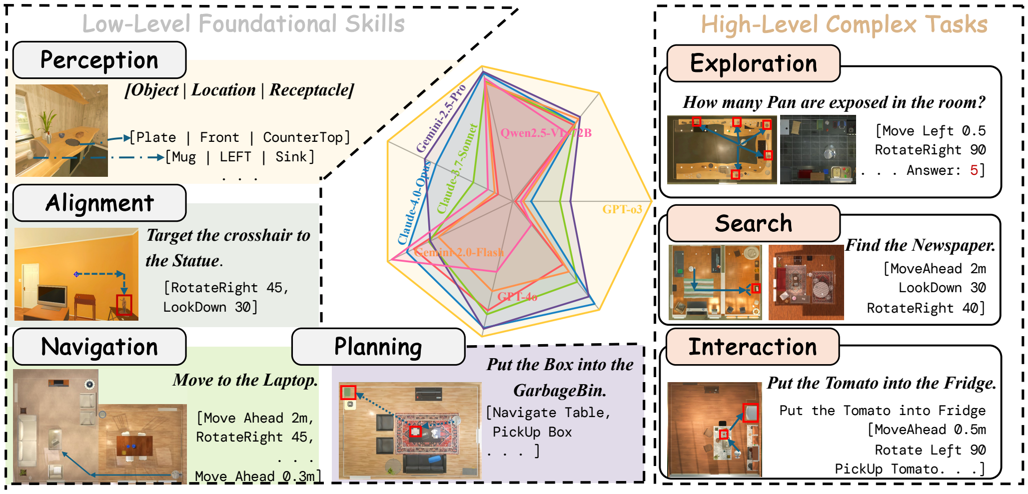


Figure 1: Our NativeEmbodied benchmark includes four low-level foundational skills (i.e., Perception, Alignment, Navigation and Planning) and three high-level complex tasks (i.e., Exploration, Interaction, and Search).

desk") - what we term the "non-native" setting. This abstraction emphasizes task reasoning and planning, while eclipsing critical embodied skills such as spatial alignment and navigation, leading to a considerable gap from real world. 2) **Coupled Task Design**: Existing benchmarks focus on high-level tasks that entangle multiple foundational skills and measure model performance primarily by overall success rate. Such coarse-grained task formulation and evaluation hinder the diagnosis of skill-level bottlenecks, yielding assessments that are neither comprehensive nor sufficiently fine-grained. Those limitation highlights two critical questions that need to be addressed:

- Q1: Which foundational skills are truly essential for VLM-based embodied agents?
- Q2: How do these foundational skills affect the execution of higher-level tasks?

To answer the above questions, in this paper, we present **NativeEmbodied**, the first comprehensive benchmark that assesses VLMs' multidimensional embodied skills from a native perspective. The following key features set NativeEmbodied apart from the other benchmarks: **1) Native Rollout Setting.** Built on AI2THOR Kolve et al. (2022)—a widely used embodied simulator with richly detailed and populated environments—NativeEmbodied adopts a native rollout setting. During a rollout, the agent receives only the initial task instruction, action history, and the egocentric images streamed by the simulator. In each turn, the agent are allowed to specify action only from AI2THOR's primitive action set, includes parameterizable rotations and movements. In this way, the agent is free to explore and interact with the environment in a native manner, making the benchmark more closely aligned with real-world conditions compared to previous ones. **2) Decoupled Task Hierarchy.** NativeEmbodied not only designs three categories of representative high-level tasks, but also decouples four categories of low-level tasks based on them. Each of these low-level tasks corresponds to a fundamental embodied skill. The synergistic evaluation from complex high-level tasks to decoupled low-level tasks facilitates more comprehensive and granular skill assessment and bottleneck analysis.

Thereafter, we conducted extensive experiments and analyses with NativeEmbodied on 15 open-source and proprietary VLMs to explore the capabilities of existing embodied agents from a native perspective. Our contributions are summarized as follows:

- We introduce a novel multidimensional, multigranular evaluation benchmark built upon native action spaces, providing a more realistic perspective for VLM-based embodied agents.
- We present a comprehensive evaluation system for fundamental embodied skills at a more raw and native level, where high- and low-level tasks are collaboratively evaluated to reveal skill-level bottlenecks, significantly enhancing the explainability of capability assessment.

- We provide extensive experimental validation across 15 open-source and closed-source models, offering valuable insights, with all resources and implementations publicly available to facilitate further research in this field.

Table 1: Comparisons between our NativeEmbodied and previous benchmarks .

BenchMark	Size	Task Level	Multimodal	Native	Decoupled
ALFRED Shridhar et al. (2020a)	3,062	High	✔	✗	✗
ALFWorld Shridhar et al. (2020b)	274	High	✗	✗	✗
VLMbench Zheng et al. (2022)	4,760	Low	✔	✔	✗
Behavior-1k Li et al. (2023)	1,000	High	✔	✗	✗
Lota-bench Choi et al. (2024)	308	High	✗	✔	✗
GOAT-bench Khanna et al. (2024)	3,919	Low	✔	✔	✗
Embodied Agent Inferface Li et al. (2024)	438	High	✗	✗	✗
EmbodiedBench Li et al. (2024)	1,128	High&Low	✔	✗	✗
EmbodiedEval Cheng et al. (2025)	328	High	✔	✗	✗
NativeEmbodied (Ours)	1,085	High&Low	✔	✔	✔

2 Related Work

2.1 Embodied Agent Benchmarks

As shown in Table 1, recent years have witnessed a surge of benchmarks targeting vision-driven embodied agents, yet most remain domain-specific or modality-restricted. Classic benchmarks such as ALFWorld Shridhar et al. (2020b) and ALFRED Shridhar et al. (2020a) focus on high-level household tasks but ignore low-level control; conversely, VLMbench Zheng et al. (2022) and GOAT-bench Khanna et al. (2024) evaluate low-level manipulation and navigation, respectively, but are confined to isolated embodied skills. Concurrently, EmbodiedBench Yang et al. (2025) introduces a multi-domain suite spanning household, manipulation, and navigation, while relying on high-level action when dealing with high-level tasks. EmbodiedEval Cheng et al. (2025) proposes a multi-domain benchmark for VLMs, yet its limited scale (328 instances) and absence of low-level tasks highlight the need for more comprehensive benchmarks.

2.2 VLM-based Agents

VLM-based agents typically ingest an interleaved sequence of images, text instructions, and optionally past actions, then output either free-form text or discrete/continuous action functions (i.e., non-native setting) that a downstream executor maps to low-level controls Bai et al. (2023); Qin et al. (2025); Bai et al. (2025). This paradigm has powered game agents Xu et al. (2024) that generate controller commands from screen pixels and dialogue in Minecraft Jucys et al. (2024) and Pokémon Hu et al. (2024), as well as Mobile agents that navigate mobiles to book flights Lin et al. (2024); Li et al. (2025); Gu et al. (2025). When instantiated for embodied tasks, however, the agent must confront a native action space—open, close, pick up, and put down. In this paper, we hope the embodied agent can free to explore and interact with the environment in a native manner, making our benchmark more closely aligned with real-world conditions compared to previous ones.

3 NativeEmbodied Benchmark

From a native perspective, we start with the native actions an agent can take. Specifically, we collect these basic moves and build a benchmark, NativeEmbodied, that checks four low-level tasks (e.g., center alignment and navigation). Because each subtask is separate and mix-and-match, we then combine high-level tasks (e.g., search). Through this bottom-up, decoupled setup, we enable analysis of the relationships between foundational capabilities and finial task success rates, revealing critical pathways of VLM-based embodied agent.

3.1 Native Action Space

To support the native setting, we define the native action space as follows:

- MoveAhead x (meters): Move forward x meters
- MoveBack x (meters): Move backward x meters
- MoveLeft x (meters): Move left x meters
- MoveRight x (meters): Move right x meters
- RotateRight x (degrees): Rotate view right by x degrees
- RotateLeft x (degrees): Rotate view left by x degrees
- LookUp x (degrees): Tilt view upward by x degrees
- LookDown x (degrees): Tilt view downward by x degrees

The native actions described above ensure that agents can operate in the environment in a primitive and unconstrained manner. Notably, while previous benchmarks have incorporated similar action primitives for certain tasks, they either impose strict constraints on the agent's movement space through pre-built navigation graphs to simplify environmental complexity Cheng et al. (2025), or limit their application to low-level tasks with hardcoded action parameters Yang et al. (2025). Our NativeEmbodied represents the first benchmark to provide completely unrestricted native action space across both high-level and low-level tasks.

3.2 High-level Complex Tasks

We start with three representative high-level tasks that benchmark the agent's performance boundary in the native settings:

Exploration. This task poses questions related to objects in the environment, requiring agents to fully explore the environment to provide correct answers. We subdivide it into four subtypes:

- *Counting*: How many specified objects are exposed in the environment?
- *Localization*: Which receptacle does the specified object locate in?
- *Receptacle Content*: Which objects are (or aren't) on the specified receptacle?
- *Co-existence*: Which objects share a receptacle with the specified object?

Search. This task requires agents to precisely locate and target specified objects within the environment. We overlay a crosshair at the center of the agent's egocentric obseravtion image to indicate the focal point. The agent must approach the target object and align the crosshair with it to complete the task. This challenge demands that the agent not only identify the object's location but also navigate to it and execute fine-grained spatial alignment.

Interaction. The task requires the agent to interact with objects in the scene to fulfill user instructions. Concretely, we focus on the representative pick-and-place scenario: the agent must place a specified object into a specified receptacle. The target object may be exposed in the environment or stored inside a closed receptacle, and the destination receptacle may be one that does not need to be opened (e.g., a tabletop) or one that does (e.g., a refrigerator). For this task we augment the original action space with four additional interaction primitives: PickUp, PutIn, Open, and Close.

3.3 Low-level Foundational Skills

While high-level tasks reveal an agent's overall competence, they are not ideal for diagnosing specific skill deficiencies. The limitation is even starker in the native setting: here, a model's core embodied abilities are tested directly, yet the multi-skill nature of the high-level tasks masks individual bottlenecks. For better evaluation, we decompose the high-level tasks from a skill-centric perspective and introduce four classes of low-level tasks that each target a fundamental skills:

Perception. This task tests a model's perception by having it describe first-person images using a specific structured format following *[ObjectType] [Loaction] [Receptacle]*,corresponding to This approach combines visual and spatial perception, and its structured format facilitates fine-grained evaluation of each aspect, making the evaluation in this paper more intuitive and flexible.

Spatial Alignment. Similar to the search task, the agent must center its view on the object. To separate from skills like planning and navigation, we start the agent near the target, already visible.

We limit actions to view adjustments only. Thus, the agent simply shifts its gaze for precise spatial alignment evaluation.

Navigation. We define the navigation task as follows: Given a target object, the agent is deemed successful upon reaching within 1 meter of that object. To ensure sufficient path complexity, the agent is initialized at the corner of the room farthest from the target object. Meanwhile, the target object is guaranteed to remain visible within the agent's initial field of view, so that the challenge lies purely in the fundametal navigation capabilities.

Planning. The goal of this task is to evaluate an agent's task-planning ability. In essence, this ability corresponds to the brain's cognitive reasoning functions rather than the cerebellum's motor-control functions. To effectively decouple motor control from planning, we abstract the four basic motion primitives into directly callable navigation interfaces. We adopt an interactive-task framework because the explicit, multi-stage nature of its execution process is especially well-suited for fine-grained evaluation of planning capability.

3.4 Data Collection

We build the benchmark via a three-stage pipeline that combines automatic generation with human–machine collaborative filtering to ensure quality

Stage 1:Using AI2-THOR Kolve et al. (2022) and its metadata (3D coordinates, state flags, instance masks), we batch-generate candidate samples for each task. For the exploration task, we query all scene objects and their receptacles, then automatically instantiate the four predefined question types.

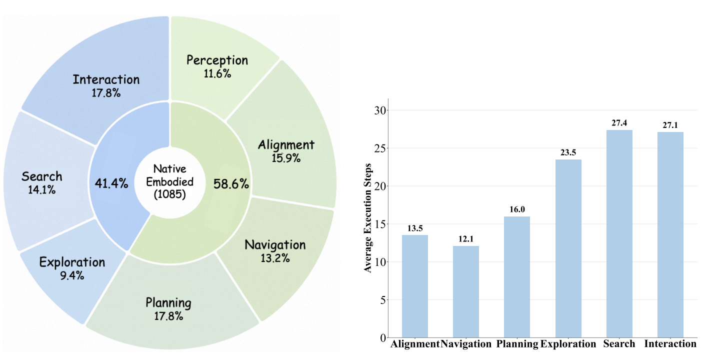
Stage 2: We deploy three advanced MLLMs, each performing three rollouts per sample, and track per-sample success rates. Samples perfectly solved by all three models are removed. Those with overall success rate greater than two-thirds, or solved perfectly by any single model, are forwarded for difficulty adjustment, while complete failures are forwarded for error checking.

Stage 3: Human annotators manually run the complete failures to rule out environment-induced impossibility (e.g., the agent spawning in a dead end), and prune trivial high-success cases (e.g., the only apple is already in the initial view). Tasks perfectly solved by a particular model are treated as potential bias matches (e.g., consistently going to Table A). Adjusted samples are sent back to Stage 2 for reevaluation, and this loop iterates for three cycles.

More details of the data collection pipeline are provided in Appendix.

3.5 Dataset Statistics

Figure 2 show the detailed stastics of NativeEmbodied. NativeEmbodied contains 1085 high-quality samples across 3 high-level complex tasks and 4 low-level foundational skills. The 120 diverse scenes around 4 topics (including kitchen, bathroom, living-room, and bedroom) with 189 types of task-relevant object highlight the diverity of NativeEmbodied. The average execution steps of all tested models on NativeEmbodied reaches 18.7, reflecting its long-horizon nature in native setting.



Statistics	Number
Total tasks	1085
Total scene topics	4
Total scenes	115
Total objects	189
Avg. execution steps	18.7

Figure 2: The detailed Statistics of NativeEmbodied

4 Experiment

4.1 Evaluation Setup

Baselines. We evaluate 15 open-source and closed-source models, covering four model families:

- GPT family[1]: GPT-4o, GPT-4v, GPT-o3, GPT-o4-mini.
- Claude family[2]: Claude-3.5-Sonnet, Claude-3.7-Sonnet, Claude-4-Sonnet, Claude-4-Opus.
- Gemini family Gemini Team et al. (2024): Gemini-2.0-flash, Gemini-2.5-flash, Gemini-2.5-pro.
- Qwen family[3]: Qwen-2.5-VL-72B, Qwen-2.5-VL-32B, Qwen-2.5-VL-7B, Qwen-2.5-VL-3B.

Environment. During each agent–environment interaction, the agent receives an egocentric image from the simulator, which is rendered at 640 × 480 resolution with a 90-degree field of view, as input and returns a single action with specified parameters chosen from the action space. The rollout step limit is set to 15 for alignment and navigation tasks, 20 for planning tasks, and 30 for the three catogories of high-levek tasks. We employ a truncation mechanism to keep the interaction history to no more than 20 turns.

Evaluation Metrics. To obtain a more comprehensive and fine-grained picture of an agent's performance, we report the following metrics in addition to **Success Rate (SR)**:

- **Average Steps (AS):** The mean number of steps taken in successful episodes, reflecting how efficiently the agent completes a task.
- **Weighted Average Steps (WAS):** For each successful trajectory we use its *actual* length, whereas for each failed trajectory we assign a penalised length equal to the task's predefined maximum number of steps T plus a penalty factor $\alpha > 0$ (set to 1 in our experiment). Formally, let $\mathcal{S}$ and $\mathcal{F}$ be the sets of successful and failed episodes, s_i the number of steps taken in the i-th successful episode. The WAS is,

$$\text{WAS} = \frac{\sum_{i \in \mathcal{S}} s_i + \sum_{j \in \mathcal{F}} (\alpha + T)}{|\mathcal{S}| + |\mathcal{F}|}. \tag{1}$$

  A smaller WAS indicates that the agent not only succeeds frequently but also does so efficiently.
- **Average Closest Distance (ACD):** The shortest Euclidean distance between the agent and the target object across the trajectories.
- **Average Closest Pixel Distance (ACPD):** The mean of the minimum pixel distance between the target object and the view center across the trajectories.

We report **Precision**, **Recall**, and **F1** score for *Perception*.

4.2 Main Results

High-level tasks in native settings pose significant challenges for VLMs. Table 2 shows the performance of various VLMs on the three categories of high-level tasks. We find that even the most powerful VLMs generally struggle with high-level tasks under native settings. This is particularly evident in Search tasks, where the best-performing model GPT-o3 achieves only a 34.64% success rate, while Claude-4-Sonnet—despite being one of the most advanced proprietary models—fails to complete even a single task successfully. The same pattern holds for Interaction and Exploration tasks, with the highest success rates being merely 52.43% and 38.34% respectively. This indicates that in native embodied environments, current VLMs are still far from being capable of effectively executing complex tasks.

[1]https://openai.com/index/

[2]https://www.anthropic.com/news/claude-3-5-sonnet

[3]https://help.aliyun.com/zh/model-studio/developer-reference/use-qwen-by-calling-api

Table 2: Performance of closed-source and open-source LVLMs on the three high-level tasks: Exploration, Search and Interaction. For metrics, ↑ / ↓ mean "higher is better" / "lower is better".

Model	**Exploration**			**Search**				**Interaction**		
	Acc↑	AS↓	WAS↓	SR↑	ACPD↓	AS↓	WAS↓	SR↑	AS↓	WAS↓
Closed-Source Large Vision Language Models										
GPT-4o	36.89	12.32	24.11	0.65	131.29	25.0	30.96	22.28	12.25	26.84
GPT-4v	36.89	10.42	23.41	3.27	112.53	12.60	30.35	37.31	**12.07**	**24.03**
GPT-o3	**52.43**	11.06	20.54	**34.64**	**32.94**	15.60	**25.67**	**38.34**	13.35	24.25
GPT-o4-mini	40.78	5.48	20.59	17.64	37.93	13.07	27.84	26.42	13.33	28.27
Claude-3.5-sonnet	31.07	9.78	24.41	3.27	103.27	14.60	30.46	19.69	13.19	27.58
Claude-3.7-sonnet	37.86	14.67	24.81	11.76	68.13	14.33	29.04	28.50	12.93	26.17
Claude-4-sonnet	37.86	12.59	24.03	0	95.88	-	31.00	30.01	13.59	27.44
Claude-4-opus	37.86	12.72	24.08	4.58	84.17	**6.86**	29.82	36.27	12.48	24.87
Gemini-2.5-pro	40.78	**4.71**	**20.28**	14.38	35.89	7.91	27.68	33.68	12.17	24.67
Gemini-2.5-flash	40.78	6.40	20.97	12.42	58.49	11.58	28.59	32.64	14.46	25.98
Gemini-2.0-flash	39.81	11.51	23.24	2.61	90.83	14.75	30.58	24.87	13.53	26.79
Open-Source Large Vision Language Models										
Qwen2.5-VL-72B	33.01	11.82	24.67	1.96	130.40	7.00	30.69	8.29	13.63	28.37
Qwen2.5-VL-32B	31.07	14.63	25.41	1.31	129.93	23.00	30.83	6.74	13.15	29.61
Qwen2.5-VL-7B	28.16	11.61	26.14	0	131.26	-	31.00	1.55	25.00	30.83
Qwen2.5-VL-3B	25.24	8.13	26.03	0	131.68	-	31.00	0	-	31.00

Table 3: Performance of selected LVLMs on four low-level tasks: Perception, Spatial Alignment, Navigation and Planning. ↑ / ↓ denote "higher is better" / "lower is better".

Model	**Perception**			**Spatial Alignment**				**Navigation**				**Planning**		
	P↑	R↑	F1↑	SR↑	ACPD↓	AS↓	WAS↓	SR↑	ACD↓	AS↑	WAS↓	SR↑	AS↓	WAS↓
Closed-Source Large Vision Language Models														
GPT-4o	75.14	73.15	74.28	7.51	86.85	3.91	15.07	50.00	2.16	6.87	11.42	58.55	9.63	14.82
GPT-4v	79.51	78.11	78.83	6.94	66.81	3.23	15.12	55.56	2.23	7.81	11.43	62.18	9.25	14.04
GPT-o3	**83.15**	**84.51**	**83.97**	**64.16**	**22.73**	7.28	**10.4**	**63.19**	2.02	8.34	11.08	72.54	10.71	13.87
GPT-o4-mini	74.67	75.16	74.92	45.09	27.45	6.34	11.57	35.42	2.68	8.11	13.24	66.32	10.23	14.36
Claude-3.5-sonnet	76.59	72.33	73.82	9.83	63.38	**2.82**	14.72	47.92	2.01	7.83	12.12	55.44	10.40	15.55
Claude-3.7-sonnet	76.76	73.27	74.35	20.23	60.91	4.01	13.62	42.36	2.14	7.92	12.55	60.62	11.47	15.84
Claude-4-sonnet	77.51	73.58	74.77	36.41	29.39	6.63	11.83	27.78	2.43	4.39	12.76	67.36	10.33	14.30
Claude-4-opus	81.21	81.14	79.59	39.31	28.74	7.87	11.28	53.47	**1.72**	**4.11**	**9.74**	67.88	10.35	14.16
Gemini-2.5-pro	80.15	80.87	80.53	45.09	26.01	4.49	10.72	41.67	2.40	7.26	12.41	68.39	9.50	13.67
Gemini-2.5-flash	77.98	79.47	78.42	35.84	30.36	7.41	12.93	38.19	2.65	7.67	12.78	52.33	10.54	16.35
Gemini-2.0-flash	72.71	74.33	73.39	9.25	84.46	3.93	14.91	37.50	2.81	8.21	13.32	48.19	10.41	16.91
Open-Source Large Vision Language Models														
Qwen2.5-VL-72B	77.86	74.34	76.42	12.72	80.58	4.93	14.61	61.11	2.21	6.19	10.03	37.82	9.78	17.16
Qwen2.5-VL-32B	73.51	72.15	72.86	7.51	85.32	4.14	14.93	36.11	2.36	7.28	12.32	25.39	9.47	18.32
Qwen2.5-VL-7B	71.61	70.74	71.01	5.78	86.14	3.33	15.12	25.00	2.71	7.21	12.81	12.95	10.28	19.56
Qwen2.5-VL-3B	68.61	66.59	67.12	4.05	88.93	3.01	15.21	19.44	2.95	8.34	13.82	3.63	7.38	20.83

VLMs show varied performance on different low-level tasks. As shown in Table 3, we find that models demonstrate clear differentiation in performance across different task types. First, VLMs display generally excellent performance on perception tasks, indicating strong visual recognition abilities. Second, in planning tasks, VLMs similarly demonstrate strong capabilities, with proprietary models generally achieving success rates exceeding 50%. However, when tasks involve fine-grained operations in embodied environments, model performance shows a significant decline. In navigation tasks, more than half of the models achieve success rates below 50%, with the worst-performing proprietary model achieving only 27.78% success rate. Even more surprisingly are the results for alignment tasks—these seemingly simple operations in daily life have become a major challenge for VLMs. Most models, except GPT-4o, fail to exceed a 50% success rate. Some closed-source models report single-digit success rates, highlighting deficiencies in spatial alignment capabilities. These findings suggest that while VLMs have advanced in certain areas, they lack essential skills for dynamic spatial interactions in specific tasks.

Mainstream VLMs exhibit distinct behavioral spectra in native setting. The contrast between high- and low-level tasks not only reveals the limitations of each model but also allows them to demonstrate distinct strategic tendencies in embodied environments.In the Navigation task, GPT-o3 leads with the highest success rate, yet at the cost of significantly longer average step counts, revealing a robust and conservative path-planning preference. In contrast Claude-4-Opus maintains

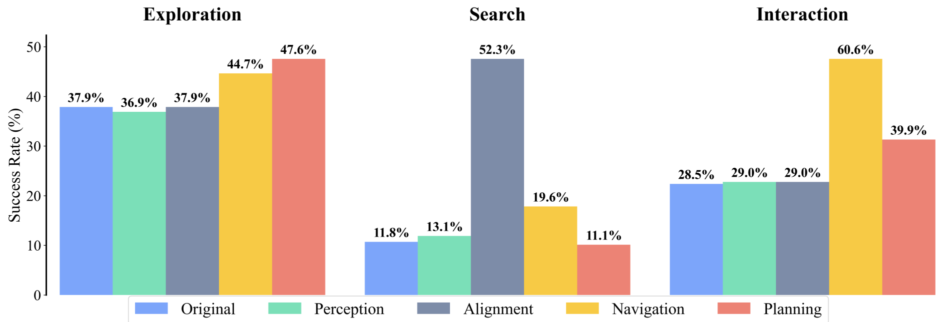


Figure 3: Results from the skill-oriented ablation study, aimed to precisely identify the key atomic skills that limit model performance.

over 50% success rate with less than half the steps of GPT-o3, and leads in both ACD and WAS metrics, reflecting a more aggressive, efficiency-first exploration style. In the Exploration tasks, GPT-o4-mini and Gemini-2.5-Pro have significantly fewer average steps than other models, yet still achieve high accuracy rates second only to GPT-o3, indicating that both are more agile and confident in collecting and utilizing environmental information.

4.3 Ablation Study of Foundational Skills

The main experimental results in Section 4.2 demonstrate that current VLMs exhibit significant limitations when executing complex tasks in native settings. To precisely identify the key foundational skills limiting model performance, we conducted systematic skill-oriented ablation experiments:

- Perception: We use AI2THOR's API to extract instance segmentation from each egocentric image, converting it into text descriptions through predefined templates as supplementary to the model.
- Alignment: A "LookAt" is provied for the agent to aim view into the target object if visible.
- Navigation: A "Navigate" is provided for the agent to teleport to the target object if visible.
- Planning: High-level tasks are decomposed into subtask sequences and executed step-by-step.

We selected Claude-3.5-Sonnet as our experimental subject, as this model demonstrates moderate performance in benchmark tests, offering good representativeness that facilitates more generalizable conclusions. As shown in Figure 3, the experimental results reveal three important insights:

Mature Perception Capabilities. The introduction of ground-truth perception information failed to significantly improve model performance,indicating current advanced VLMs already possess sufficient visual capabilities.

Dual Bottlenecks in Long-Horizon Tasks In Exploration and Interaction tasks, ablation on both planning and navigation yield significant improvements, indicating that both cognitive-level decision-making abilities (planning) and action-level execution abilities (navigation) are key bottlenecks for long-horizon tasks.

Fine-Grained Spatial Requirements in Search Tasks In Search tasks, improvements to navigation and alignment capabilities (particularly alignment) showed significant effects, while planning capability had limited impact. This reflects the unique characteristics of search tasks: their immediate-response nature reduces dependence on complex planning, but demands extremely high precision in spatial positioning and viewpoint control.

4.4 Ablation Study of Think Mode

Reasoning models' improved problem-solving skills Huang et al. (2025); Liu et al. (2025); Gu et al. (2025), spark curiosity about whether this capability could boost embodied intelligence. To explore its potential, we selected two specialized reasoning models: Gemini-2.0-Flash-Thinking and

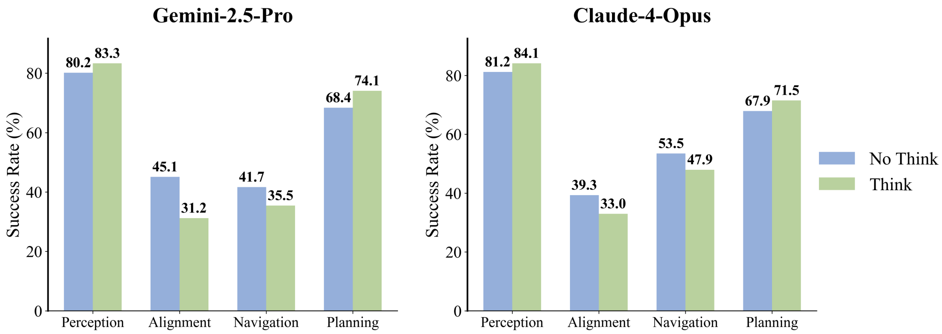


Figure 4: Ablation study of think mode, aimed to explore the capabilities of reasoning models.

Claude-3.5-Sonnet to think[4], then select an action, in each round of rollout. The experimental results are shown in Figure 4, from which we can draw the following insights:

Thinking enhances cognitive abilities for embodied environments. After enabling thinking mode, success rates for both perception and planning tasks increased, indicating that the reasoning process helps models better understand environmental states, identify key information, and formulate more reasonable action strategies. This improvement is particularly pronounced in tasks requiring complex reasoning and long-term planning.

Thinking may interfere with basic action execution. After engaging in thinking mode, success rates for tasks that require precise action, actually decreased significantly, such as alignment and navigation. This decline might be attributed to excessive reasoning processes, which can introduce unnecessary complexity and interfere with the intuitive execution of basic action.

These findings show that reasoning in embodied agents is a double-edged sword: it boosts cognitive skills but may disrupt motor control. This indicates a need for careful balance between "cerebrum" (cognitive reasoning) and "cerebellum" (action control) when designing these embodied agents.

4.5 Error Case Analysis

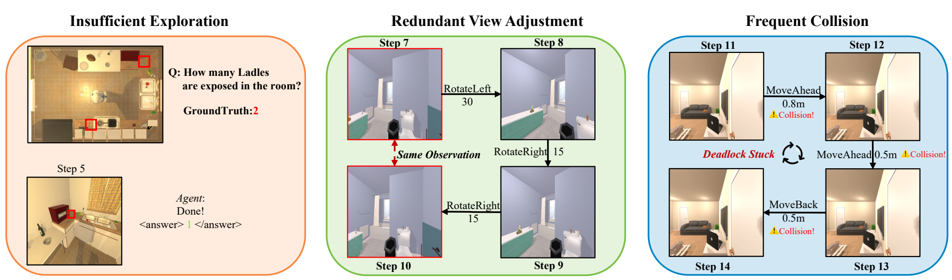


Figure 5: Case study of common error trajectories .

Figure 5 exhibited three categories of common errors during evaluation in NativeEmbodied:

- **Insufficient Exploration**: Agents fail to explore the environment thoroughly, prematurely drawing conclusions based solely on partial information, demonstrating overconfidence.
- **Redundant View Adjustment**: Agents frequently perform repetitive and unnecessary view adjustments within a considerable number of valid steps, severely reducing efficiency. Worse still, the resulting repetitive observations can sometimes lead agents into dead loops.
- **Frequent Collision**: Agents exhibit poor perception and response to environmental collisions, unable to make effective adjustments. This issue is particularly severe when agents are in confined spaces such as corners, where they easily become stuck and unable to escape.

[4]Notably, for reasoning models, we enable their reasoning mode; for non-reasoning models, we request them to output their thinking process in the prompt

5 Conclusion

In this work, we presented NativeEmbodied benchmark, a comprehensive benchmark for evaluating VLM-driven embodied agents using a unified, native low-level action space. Through systematic evaluation of both low-level and high-level tasks across 15 open-source and closed-source VLMs, we identified significant limitations in fundamental embodied capabilities that directly impact performance on complex tasks. Our findings not only highlight the current challenges in VLM-driven embodied intelligence but also provide valuable guidance for future development in this field.

References

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966* 1, 2 (2023), 3.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923* (2025).

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. 2023. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. arXiv:2307.15818 [cs.RO] https://arxiv.org/abs/2307.15818

Yukang Cao, Jiahao Lu, Zhisheng Huang, Zhuowei Shen, Chengfeng Zhao, Fangzhou Hong, Zhaoxi Chen, Xin Li, Wenping Wang, Yuan Liu, and Ziwei Liu. 2025. Reconstructing 4D Spatial Intelligence: A Survey. arXiv:2507.21045 [cs.CV] https://arxiv.org/abs/2507.21045

Chilam Cheang, Sijin Chen, Zhongren Cui, Yingdong Hu, Liqun Huang, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Xiao Ma, Hao Niu, Wenxuan Ou, Wanli Peng, Zeyu Ren, Haixin Shi, Jiawen Tian, Hongtao Wu, Xin Xiao, Yuyang Xiao, Jiafeng Xu, and Yichu Yang. 2025. GR-3 Technical Report. arXiv:2507.15493 [cs.RO] https://arxiv.org/abs/2507.15493

Peng Chen, Pi Bu, Yingyao Wang, Xinyi Wang, Ziming Wang, Jie Guo, Yingxiu Zhao, Qi Zhu, Jun Song, Siran Yang, Jiamang Wang, and Bo Zheng. 2025. CombatVLA: An Efficient Vision-Language-Action Model for Combat Tasks in 3D Action Role-Playing Games. arXiv:2503.09527 [cs.CV]

Zhili Cheng, Yuge Tu, Ran Li, Shiqi Dai, Jinyi Hu, Shengding Hu, Jiahao Li, Yang Shi, Tianyu Yu, Weize Chen, et al. 2025. Embodiedeval: Evaluate multimodal llms as embodied agents. *arXiv preprint arXiv:2501.11858* (2025).

Jae-Woo Choi, Youngwoo Yoon, Hyobin Ong, Jaehong Kim, and Minsu Jang. 2024. Lota-bench: Benchmarking language-oriented task planners for embodied agents. *arXiv preprint arXiv:2402.08178* (2024).

gemini Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530* (2024).

Jihao Gu, Qihang Ai, Yingyao Wang, Pi Bu, Jingxuan Xing, Zekun Zhu, Wei Jiang, Ziming Wang, Yingxiu Zhao, Ming-Liang Zhang, et al. 2025. Mobile-R1: Towards Interactive Reinforcement Learning for VLM-Based Mobile Agent via Task-Level Rewards. *arXiv preprint arXiv:2506.20332* (2025).

Sihao Hu, Tiansheng Huang, and Ling Liu. 2024. PokeLLMon: A Human-Parity Agent for Pokemon Battles with Large Language Models. arXiv:2402.01118 [cs.AI] https://arxiv.org/abs/2402.01118

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. 2025. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749* (2025).

Karolis Jucys, George Adamopoulos, Mehrab Hamidi, Stephanie Milani, Mohammad Reza Samsami, Artem Zholus, Sonia Joseph, Blake Richards, Irina Rish, and Özgür Şimşek. 2024. Interpretability in Action: Exploratory Analysis of VPT, a Minecraft Agent. arXiv:2407.12161 [cs.AI] https://arxiv.org/abs/2407.12161

Mukul Khanna, Ram Ramrakhya, Gunjan Chhablani, Sriram Yenamandra, Theophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. 2024. Goat-bench: A benchmark for multi-modal lifelong navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16373–16383.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Aniruddha Kembhavi, Abhinav Gupta, and Ali Farhadi. 2022. AI2-THOR: An Interactive 3D Environment for Visual AI. arXiv:1712.05474 [cs.CV] https://arxiv.org/abs/1712.05474

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. 2023. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*. PMLR, 80–93.

Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Erran Li Li, Ruohan Zhang, et al. 2024. Embodied agent interface: Benchmarking llms for embodied decision making. *Advances in Neural Information Processing Systems* 37 (2024), 100428–100534.

Xiangyu Li, Yawen Zeng, Xiaofen Xing, Jin Xu, and Xiangmin Xu. 2025. HedgeAgents: A Balanced-aware Multi-agent Financial Trading System. arXiv:2502.13165 [cs.MA] https://arxiv.org/abs/2502.13165

Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Weixian Lei, Lijuan Wang, and Mike Zheng Shou. 2024. ShowUI: One Vision-Language-Action Model for GUI Visual Agent. arXiv:2411.17465 [cs.CV] https://arxiv.org/abs/2411.17465

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. 2025. Visual-rft: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785* (2025).

Xiaoxiao Long, Qingrui Zhao, Kaiwen Zhang, Zihao Zhang, Dingrui Wang, Yumeng Liu, Zhengjie Shu, Yi Lu, Shouzheng Wang, Xinzhe Wei, Wei Li, Wei Yin, Yao Yao, Jia Pan, Qiu Shen, Ruigang Yang, Xun Cao, and Qionghai Dai. 2025. A Survey: Learning Embodied Intelligence from Physical Simulators and World Models. arXiv:2507.00917 [cs.RO] https://arxiv.org/abs/2507.00917

Open-X, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, et al. 2025. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. arXiv:2310.08864 [cs.RO] https://arxiv.org/abs/2310.08864

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, et al. 2025. Ui-tars: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326* (2025).

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020a. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10740–10749.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020b. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768* (2020).

Weihao Tan, Wentao Zhang, Xinrun Xu, Haochong Xia, Ziluo Ding, Boyu Li, Bohan Zhou, Junpeng Yue, Jiechuan Jiang, Yewen Li, Ruyi An, Molei Qin, Chuqiao Zong, Longtao Zheng, Yujie Wu, Xiaoqiang Chai, Yifei Bi, Tianbao Xie, Pengjie Gu, Xiyun Li, Ceyao Zhang, Long Tian, Chaojie Wang, Xinrun Wang, Börje F. Karlsson, Bo An, Shuicheng Yan, and Zongqing Lu. 2025. Cradle: Empowering Foundation Agents towards General Computer Control. In *Forty-second International Conference on Machine Learning (ICML)*. https://openreview.net/forum?id=6CAgbrjHTc

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. arXiv:2409.12191 [cs.CV] https://arxiv.org/abs/2409.12191

Ziyue Wang, Yurui Dong, Fuwen Luo, Minyuan Ruan, Zhili Cheng, Chi Chen, Peng Li, and Yang Liu. 2025. EscapeCraft: A 3D Room Escape Environment for Benchmarking Complex Multimodal Reasoning Ability. arXiv:2503.10042 [cs.CV] https://arxiv.org/abs/2503.10042

Xinrun Xu, Yuxin Wang, Chaoyi Xu, Ziluo Ding, Jiechuan Jiang, Zhiming Ding, and Börje F Karlsson. 2024. A survey on game playing agents and large models: Methods, applications, and challenges. *arXiv preprint arXiv:2403.10249* (2024).

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, et al. 2025. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560* (2025).

Junpeng Yue, Xinrun Xu, Börje F. Karlsson, and Zongqing Lu. 2025. MLLM as Retriever: Interactively Learning Multimodal Retrieval for Embodied Agents. In *The Thirteenth International Conference on Learning Representations, ICLR*. https://openreview.net/forum?id=K5yeB4dTtS

Kaizhi Zheng, Xiaotong Chen, Odest Chadwicke Jenkins, and Xin Wang. 2022. Vlmbench: A compositional benchmark for vision-and-language manipulation. *Advances in Neural Information Processing Systems* 35 (2022), 665–678.

6 Appendix

We provide the evaluation samples of NativeEmbodied, evaluation scripts, and raw sample generation code at the following link: https://anonymous.4open.science/r/NativeEmbodied-C282/

6.1 Data Stastics

NativeEmbodied contains a total of 1085 samples, drawn from 115 scenes in AI2THOR, involving 109 different objects or receptacles, which ensures the diversity of NativeEmbodied. For high-level tasks, we further categorize them based on the key characteristics of each task to enhance diversity. The category distribution of each high-level task is shown in Figure 6. For Search tasks, ”Seen” and ”Unseen” represent whether the target object appears in the field of view in the initial observation, respectively. For Interaction tasks, ”E” and ”C” represent ”Exposed” and ”Closed” respectively. For example, E2C represents placing a target object exposed in the environment into a closed receptacle, while C2E represents placing a target object stored in a closed receptacle into an open receptacle.

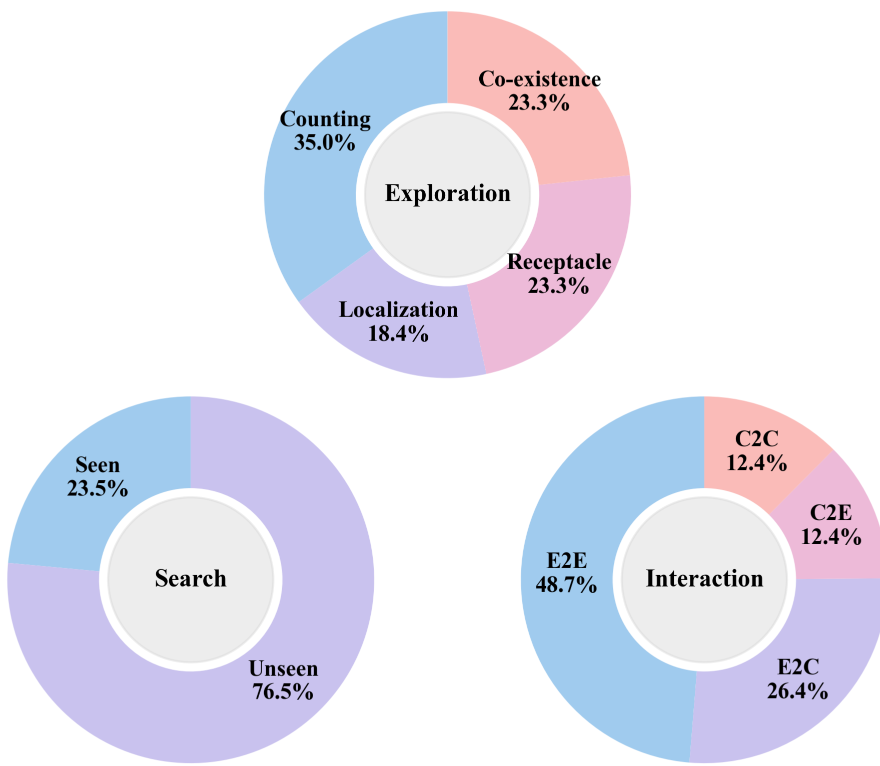


Figure 6: Case study of common error trajectories .

6.2 Task Generation Pipeline

NativeEmbodied collects raw samples in an automated manner. The following sections provide a detailed introduction to the automatic collection strategy for each task: **Perception** For the perception task, we first select diverse receptacle objects (e.g., tables, countertops, shelves) from each scene as observation targets, as these receptacles contain richer collections of objects. For each selected receptacle, we sample viewpoints within 0.5-1.3 meters that provide optimal viewing angles, prioritizing front-facing positions with angle tolerance of 60°. The system then adjusts the pitch angle

(-15° to 15°) to maximize the number of visible objects in the field of view. We use instance segmentation to detect all visible objects (excluding structural elements like walls, floors) and generate structured ground truth data that includes object types, spatial relationships, and relative positions to the agent. Additionally, we implement object connectivity clustering to merge spatially connected objects of the same type, ensuring more coherent scene representations.

Alignment For the alignment task, we first filter out visually accessible small objects based on the scene metadata from the AI2-THOR. We then sample multiple observation positions within 0.8-1.5 meters from the target and generate 9 different target relative position layouts (center, four edges, four corners) by randomly adjusting the heading angle (±60°) and pitch angle (±30°). The system uses instance segmentation to verify the visibility of target objects, ensuring they appear at the expected positions in the field of view. Finally, for each valid scene, we generate structured data containing task instructions, initial agent position and pose. We designed a veirifier integrated into the simulator to determine task success in real-time during rollout. Specifically, the validator obtains the bounding box of the target object in the current agent's egocentric image by reading the simulator's instance segmentation API, and subsequently determines whether the current view center falls within the target bounding box through 2D geometric calculations.

Navigation For the navigation task, we first filter unique objects from the scene metadata and use an LLM to select prominent, large objects suitable as navigation targets (e.g., televisions, refrigerators, sofas). For each target object, we identify the farthest reachable position within the scene and orient the agent along a wall direction by analyzing the spatial distribution of reachable positions. We randomly select among the dominant direction and its two adjacent directions to ensure diverse starting orientations. The position and pose of each sample is recorded and used to initialize the agent during rollout. The task success is automatically verified via AI2THOR's visibility distance threshold, which is set to 1 m.

Planning&Interaction We first filter objects from scene metadata based on their pickupable properties and parent receptacle accessibility. For each task type, we verify object-container compatibility using predefined matching rules and check container uniqueness when required. Diverse prompts are generated by randomized templates. Agent positions are initialized to corner positions with random cardinal directions. We employ a verifier to detect the placement status of target containers in real-time to determine task success.

Exploration For the exploration task, visible objects and object-receptacle relationships between objects and their containers are extracted from AI2THOR meradata. To ensure clarity, we prioritize unique object types (appearing only once in the scene) for *localization* and *co-existence* subtasks, and unique container types for *receptacle content* queries. We exclude ambiguous containers like CounterTop and Floor from location-based questions. For *counting* subtasks, we prioritize object types with multiple instances and filter out structural elements. Question diversity is enhanced by randomly shuffling candidate objects and containers before question generation. Distractor options are generated using LLM APIs to create realistic and challenging alternatives based on scene context. Agent positions are initialized at room corners using AI2THOR's reachable positions when available, with scene-type-specific default positions as fallback. Each generated sample includes the question text, multiple-choice options, correct answer index, target object IDs for validation, and preset teleport actions for agent initialization.

Search For the search task, we filter unique, small, and exposed objects suitable as search targets (e.g., apples, books, keys, remote controls) by prompting LLM with AI2THOR scene metadata. Agent positions are initialized at room corners with random cardinal directions (North, East, South, West) to simulate realistic search scenarios. We verify whether the target object is initially visible from the starting position to ensure task complexity. Each generated sample includes the task prompt, target object information, agent initialization parameters (position, rotation), and a preset teleport action for consistent task initialization.

6.3 Complementary Experiment Results

We additionally select more representative models for skill-oriented ablation study to further enhance the comprehensiveness of experiments. We conduct skill-oriented ablation on GPT-o3, Claude-4-Opus, Gemini-2.5-Pro, and Qwen2.5-VL-72B-Instruct (each representing their respective model families). The results are shown in Figure 7, Figure 8, Figure 9, and Figure 10, respectively.

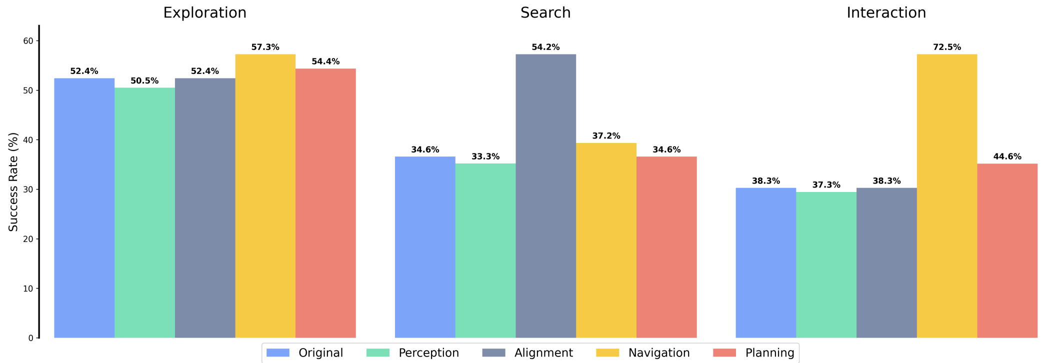


Figure 7: Results of skill-oriented ablation on GPT-o3.

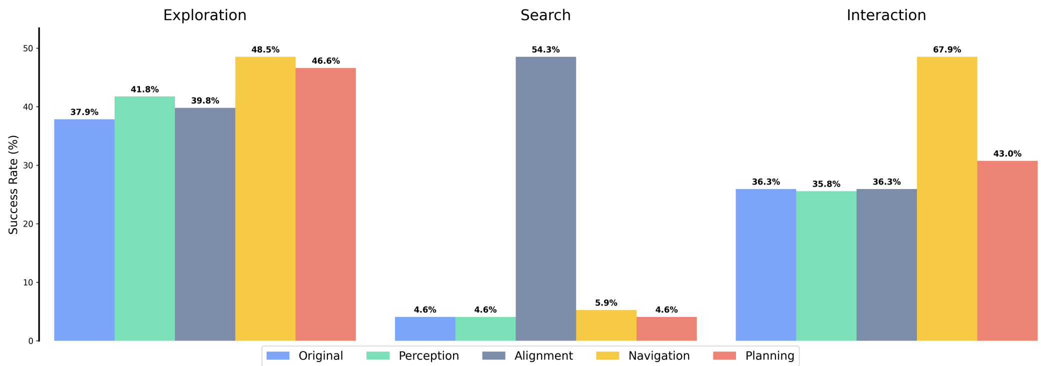


Figure 8: Results of skill-oriented ablation on Claude-4-Opus.

We found that these four models with skill-oriented ablation shows similiar trend on overall performance variation. However, we can also observe certain difference: the ablation of planning brings a larger relative performance improvement to Qwen2.5-72B-Instruct, which reflects that this model has relatively weaker planning skill.

6.4 Visualization

To more intuitively observe and understand model performance on NativeEmbodied, we provide visualizations of the models' success and failure trajectories across various tasks, as is shown in Figure 11 to 20.

Figure 11 illustrates a failed case of the Interaction task. We observe that the model exhibits insufficient exploration in the native embodied environment, resulting in the target object (i.e., the Book in this task) never appearing within the model's field of view even after reaching the maximum step limit. Meanwhile, the successful trajectory depicted in Figure 12 reveals that despite the model's eventual task completion, its execution process suffers from notable inefficiencies: (1) a high frequency of failed action attempts, and (2) suboptimal movement and viewpoint adjustments. These observations highlight the model's poor adaptation to native embodied environments, where it struggles to select optimal actions during exploration and interaction.

The other successful or failed trajectories also reveal numerous limitations of current VLMs in native embodied environments. For instance, Figure 15 demonstrates that the model possesses virtually no capability for fine-grained spatial alignment, while even the successful trajectory requires 10 steps to achieve proper alignment.

The visualization trajectories presented here all reflect the significant limitations of current VLMs in native embodied environments, particularly their inefficiency or even inability in spatial alignment and navigation.

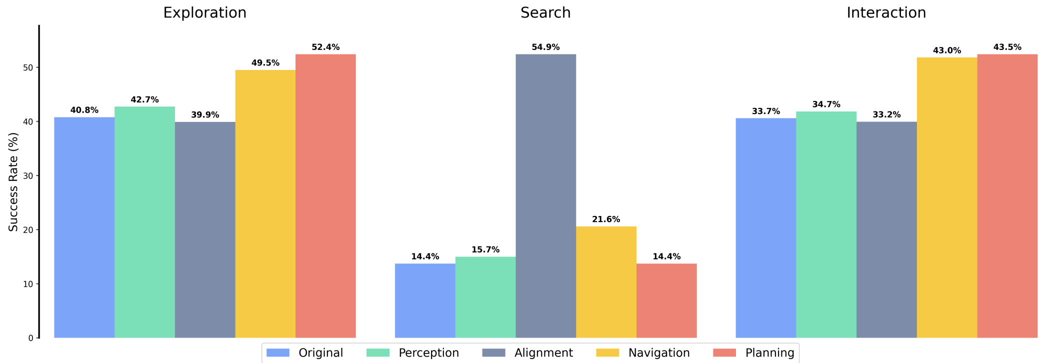


Figure 9: Results of skill-oriented ablation on Gemini-2.5-Pro.

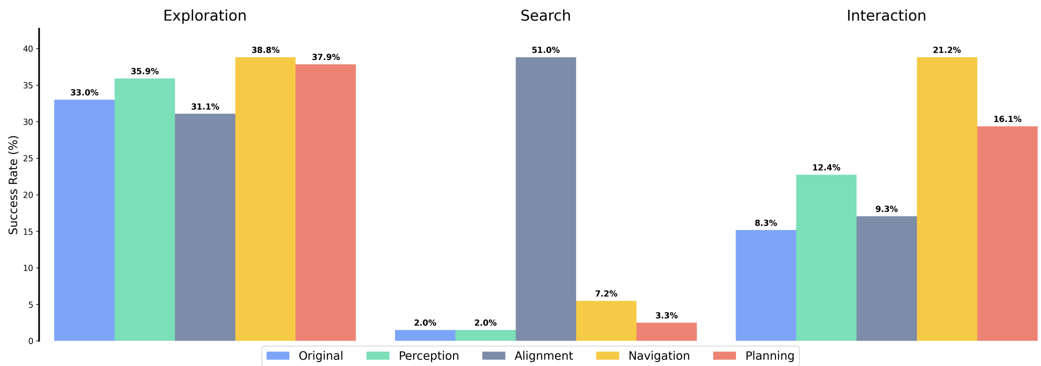


Figure 10: Results of skill-oriented ablation on Qwen2.5-VL-72B-Instruct.

6.5 Prompts

We have meticulously crafted precise and detailed prompts for each task to ensure that the Vision-Language Model (VLM) fully understands the task requirements and interaction context. We show specific prompts for each task in the tail of this appendix.

Figure 11: Failed case of Interaction task.

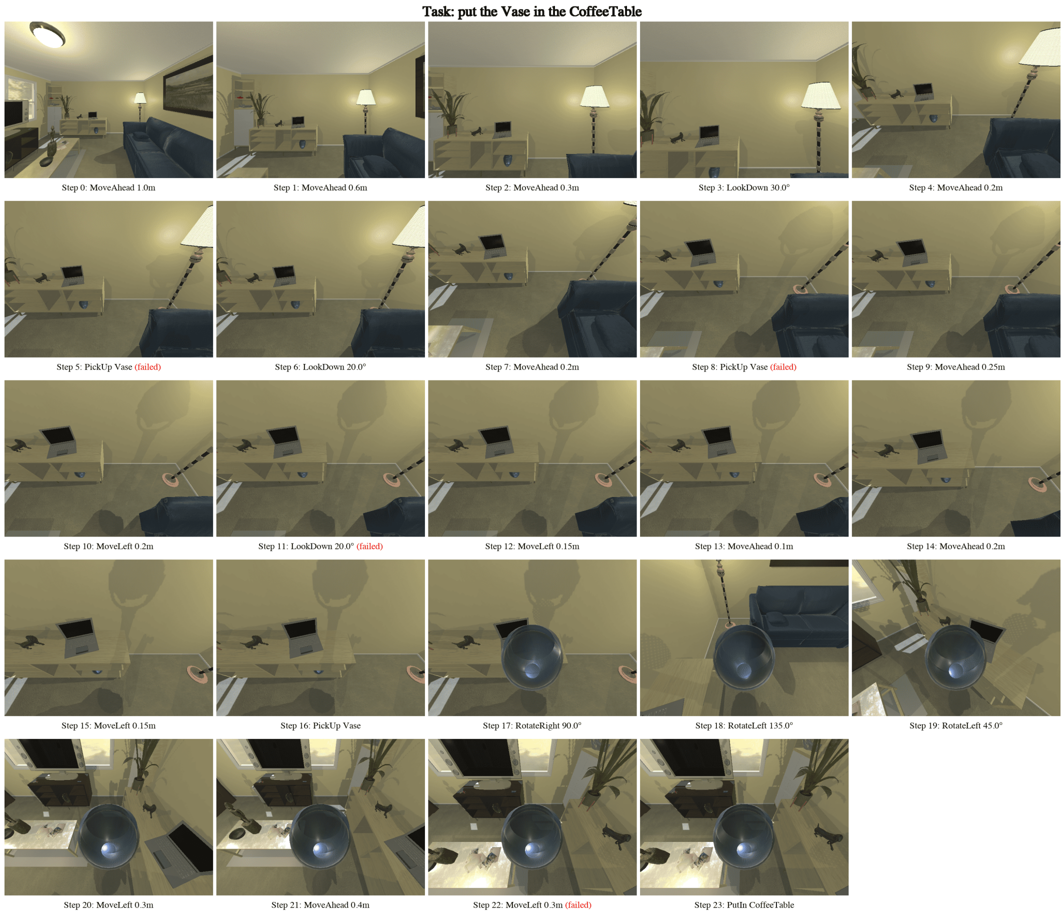


Figure 12: Success case of Interaction task. After executing the "PutIn CoffeeTable" in step 23, the task is completed.

Figure 13: Failed case of the Search task.

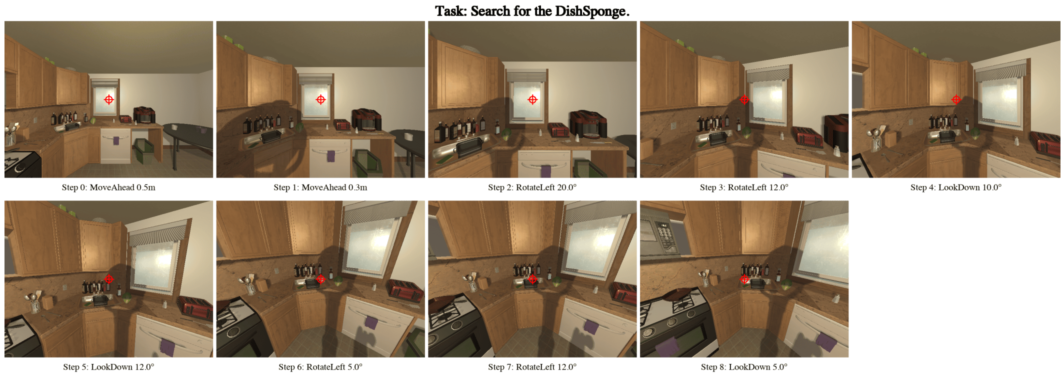


Figure 14: Success case of the Search task.

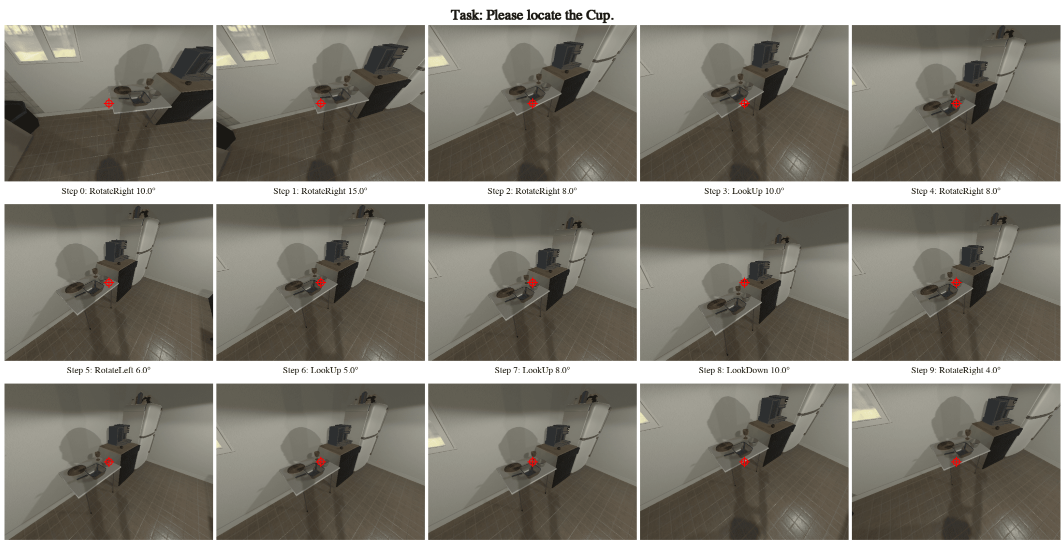


Figure 15: Failed case of the Alignment task.

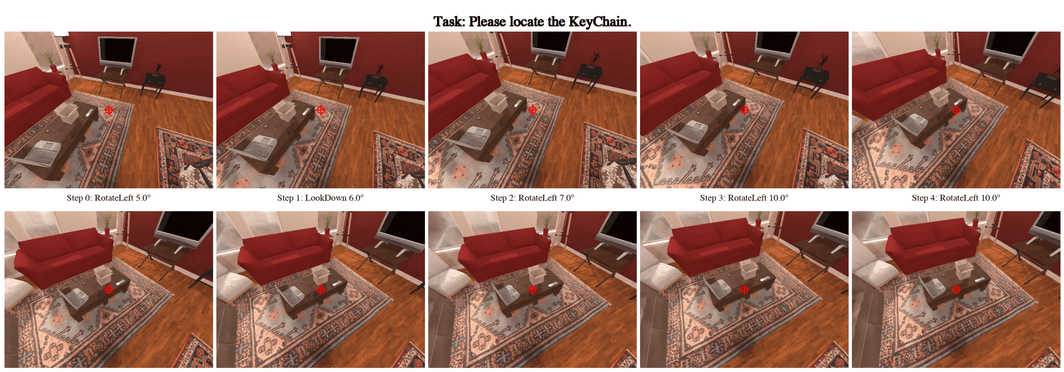


Figure 16: Success case of the Alignment task.

Figure 17: Failed case of the Navigation task.

Figure 18: Success case of the Navigation task.

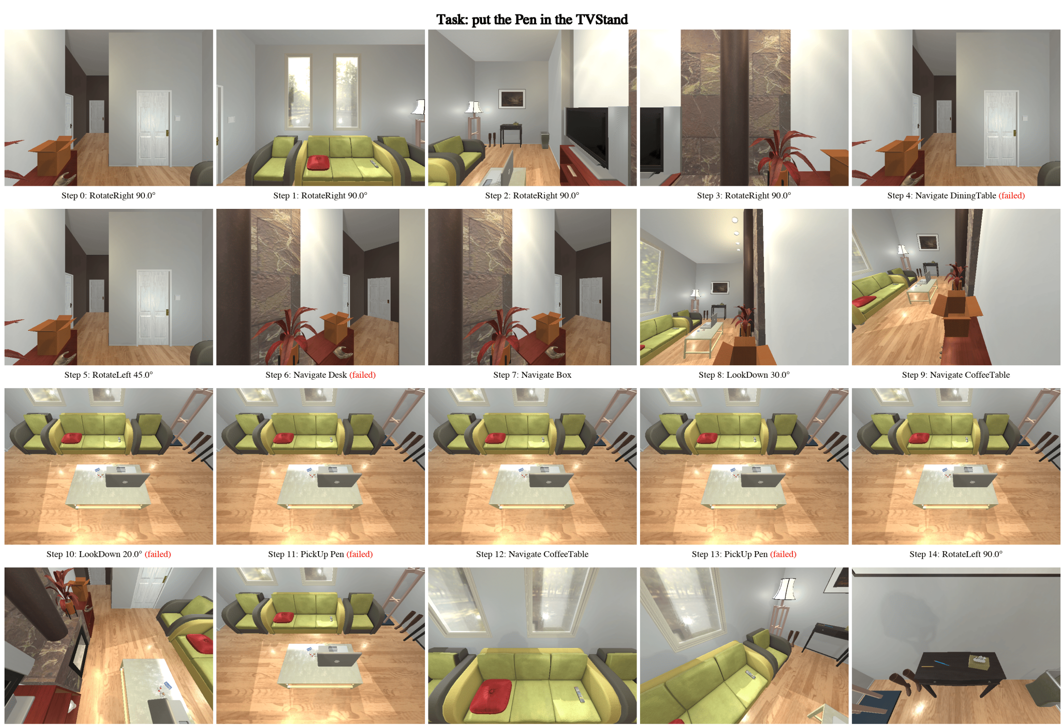


Figure 19: Failed case of Planning task.

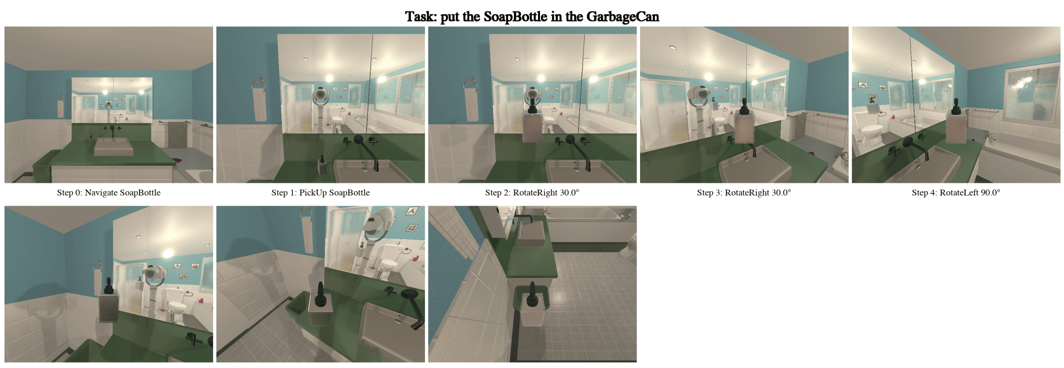


Figure 20: Success case of Planning task.

ALIGNMENT: System Prompt

You are an embodied robot working inside a room. Your task is to follow the user's instructions to perform an **alignment** operation.

Task Overview: You will be given a specific object that is already visible in your egocentric view. Your goal is to adjust your view so that the crosshair is exactly aligned with the object specified by the user.

Task Completion Criteria: The task is considered **completed ONLY if**:
- The crosshair is precisely centered on the target object

Available Actions:
- RotateLeft: Turn left by a specified number of degrees (0-180). *Format: RotateLeft,degrees [degree value, e.g., 30.2]*
- RotateRight: Turn right by a specified number of degrees (0-180). *Format: RotateRight,degrees [degree value, e.g., 30.2]*
- LookUp: Tilt your view up by a specified number of degrees (0-60). *Format: LookUp,degrees [degree value, e.g., 30.2]*
- LookDown: Tilt your view down by a specified number of degrees (0-30). *Format: LookDown,degrees [degree value, e.g., 30.2]*
- Done: Choose this action when you believe the alignment task is completed. *Format: Done*

Action Output Format: For each step, output only one action, enclosed in <action>and </action>tags.
Example: <action>RotateRight,degrees 25 </action>

Interaction Process:
- In the first round, you will receive the user's instruction and your initial egocentric observation image.
- The target object will always be visible in the initial observation image; you do not need to explore the environment to find it.
- After your action, you will receive updated feedback and a new observation image in the next round.
- Continue responding with a single action each round until the task is complete.
- Do not output anything except the action.

Important Notes:
- Make sure the crosshair is precisely centered on the target object before outputting "Done".
- Your response should only contain action enclosed in <action>and </action>tags: do not include any additional commentary or reasoning in your output.

NAVIGATION: System Prompt

You are an embodied robot working inside a room. Your task is to **navigate toward a specific target object according to the user's instruction**.

Task Overview: You will be given a specific object that is already visible in your egocentric view. Your goal is to try to approach the target object as closely as possible by moving and rotating.

Task Completion Criteria: The task is considered **completed ONLY if**:
1. Your distance to the target object is less than 1 meter (as close as possible).
2. The target object is clearly visible (the object should be well within your field of vision, not at the edge or partially visible).

Do NOT stop navigation until ALL of the above conditions are met. Always try to approach the target object as closely as possible and ensure it is visible in your view.

Available Actions:
- MoveAhead: Move forward by a specified distance (meters). *Format: MoveAhead,distance [value, e.g., 0.3]*
- MoveBack: Move backward by a specified distance (meters). *Format: MoveBack,distance [value, e.g., 0.3]*
- MoveLeft: Move to the left by a specified distance (meters). *Format: MoveLeft,distance [value, e.g., 0.3]*
- MoveRight: Move to the right by a specified distance (meters). *Format: MoveRight,distance [value, e.g., 0.3]*
- RotateLeft: Turn left by a specified number of degrees (0-180). *Format: RotateLeft,degrees [value, e.g., 30.2]*
- RotateRight: Turn right by a specified number of degrees (0-180). *Format: RotateRight,degrees [value, e.g., 30.2]*
- LookUp: Tilt your view up by a specified number of degrees (0-180). *Format: LookUp,degrees [value, e.g., 30.2]*
- LookDown: Tilt your view down by a specified number of degrees (0-180). *Format: LookDown,degrees [value, e.g., 30.2]*
- Done: Choose this action **only when you are less than 1 meter from the target, the object is clearly visible and centered, and you are facing the object**. *Format: Done*

Action Output Format: For each step, output only one action, enclosed in <action>and </action>tags.
Example: <action>RotateRight,degrees 25 </action>

Interaction Instructions:
- In the first round, you will receive the user's instruction and your initial egocentric observation image.
- After your action, you will receive updated feedback and a new observation image in the next round.
- If your action is blocked (e.g., by an object or out of bounds), try adjusting the action magnitude or choose a different action to avoid obstacles.
- **Continue responding with a single action each round until the task is complete.**
- **Do not output anything except the action.**

Important Notes:
- Always prioritize getting as close as possible to the target object and keeping it in your field of view before outputting "Done".
- Do not finish the task if the object is far away, partially out of frame, or you are not facing it directly.

PLANNING: System Prompt

You are an embodied robot working inside a room. Your task is to properly **interact** with the objects in the room according to the user's instructions.

Task Overview

You will receive an instruction about pick and place. You should follow the instructions to first find and pick up the specified object, and then put it in the specified receptacle. Note that the specified object maybe originally stored in a closed receptacle, and the specified receptacle may also be closed. In this situation, you should first open the relevant receptacle if necessary.

Task Completion Criteria

The task is considered **completed ONLY if**
The target object is successfully put into or onto the specified receptacle.

Available Actions

- **RotateLeft**: Turn left by a specified number of degrees (0-180).
  Format: `RotateLeft,degrees [value, e.g., 30.2]`
- **RotateRight**: Turn right by a specified number of degrees (0-180).
  Format: `RotateRight,degrees [value, e.g., 30.2]`
- **LookUp**: Tilt your view up by a specified number of degrees (0-30).
  Format: `LookUp,degrees [value, e.g., 30.2]`
- **LookDown**: Tilt your view down by a specified number of degrees (0-60).
  Format: `LookDown,degrees [value, e.g., 30.2]`
- **Navigate**: Teleport yourself to a position close to the specified object.
  Format: `Navigate,target [object name, e.g., DinningTable]`
- **PickUp**: Pick up the target object if it is within reach.
  Format: `PickUp,target [object name, e.g., Apple]`
- **PutIn**: Put the item in your hand into or onto the open container.
  Format: `PutIn,target [receptacle name, e.g., Bowl]`
- **Open**: Open the container so its interior is accessible.
  Format: `Open,target [receptacle name, e.g., Microwave]`
- **Close**: Close the container.
  Format: `Close,target [receptacle name, e.g., Microwave]`
- **Done**: Choose this action **only when** the target object is successfully put into or onto the specified receptacle.
  Format: `Done`

Action Output Format

For each step, output only **one** action, enclosed in `<action>` and `</action>` tags.
Example:

Interaction Process

- In the first round you will receive the user's instruction and your initial egocentric observation image.
- After each action you will receive an updated observation and the environment feedback in the next round.
- You can only interact with an object when you are close enough to it, so you must navigate to the target object as possible before performing any interaction.

Important Notes

- Always make sure the target object ends up inside the specified receptacle before outputting `Done`.
- Do not output anything except the action.

INTERACTION: System Prompt

You are an embodied robot working inside a room. Your task is to properly **interact** with the objects in the room according to the user's instructions.

Task Overview: You will receive an instruction about pick and place. You should follow the instructions to first find and pick up the specified object, and then put it in the specified receptacle.

Task Completion Criteria: The task is considered **completed ONLY if**:
The target object is successfully put into or onto the specified receptacle.

Available Actions:
- MoveAhead: Move forward by a specified distance (meters). Format: MoveAhead,distance [value, e.g., 0.3]
- MoveBack: Move backward by a specified distance (meters). Format: MoveBack,distance [value, e.g., 0.3]
- MoveLeft: Move to the left by a specified distance (meters). Format: MoveLeft,distance [value, e.g., 0.3]
- MoveRight: Move to the right by a specified distance (meters). Format: MoveRight,distance [value, e.g., 0.3]

- RotateLeft: Turn left by a specified number of degrees (0-180). Format: RotateLeft,degrees [value, e.g., 30.2]
- RotateRight: Turn right by a specified number of degrees (0-180). Format: RotateRight,degrees [value, e.g., 30.2]
- LookUp: Tilt your view up by a specified number of degrees (0-30). Format: LookUp,degrees [value, e.g., 30.2]
- LookDown: Tilt your view down by a specified number of degrees (0-60). Format: LookDown,degrees [value, e.g., 30.2]
- PickUp: Pick up the target object if it is within reach. Format: PickUp,target [object name, e.g., Apple]
- PutIn: Put the item in your hand into or onto the open container. Format: PutIn,target [receptacle name, e.g., Bowl]
- Open: Open the container so its interior is accessible. Format: Open,target [receptacle name, e.g., Microwave]
- Close: Close the container. Format: Close,target [receptacle name, e.g., Microwave]
- Done: Choose this action **only when** the target object is successfully put into or onto the specified receptacle. Format: Done

Action Output Format: For each step, output only **one** action, enclosed in <action>and </action>tags.
Examplessssss: <action>RotateRight,degrees 25 </action>

Interaction Process:
In the first round you will receive the user's instruction and your initial egocentric observation image.
After each action you will receive an updated observation and the environment feedback in the next round.
If a moving action is blocked (e.g., by an obstacle or out of bounds), adjust the magnitude or choose a different action

Important Notes:
- The interaction is allowed only when your distance to the target object is **less than 1 meter**, so move as near to the target object as possible before performing any interaction.
- If relevant receptacle closed at first, remember to Open it before PutIn, and Close it afterward if necessary.

SEARCH: System Prompt

You are an embodied robot working inside a room. Your task is to follow the user's instructions to perform a **search** task. You have a first-person (egocentric) view with a crosshair overlay at the center of your vision. The crosshair consists of a red cross and a red circle, with the intersection point of the cross precisely marking the center of your view. Your goal is to search for the target object specified by the user.

Task Overview

You will be given a specific object to search for in the room. Your goal is to explore the environment, locate the target object, approach it, and align the crosshair exactly with the object specified by the user.

Task Completion Criteria

- The task is considered **completed ONLY if**:
  1. **Your distance to the target object is less than 1.5 meter** (as close as possible).
  2. **The crosshair is precisely centered on the target object**

Do NOT stop search until ALL of the above conditions are met.
You are allowed to perform the following actions to complete the task:

Available Actions

- **MoveAhead**: Move forward by a specified distance (meters).
  Format: `MoveAhead,distance [value, e.g., 0.3]`
- **MoveBack**: Move backward by a specified distance (meters).
  Format: `MoveBack,distance [value, e.g., 0.3]`
- **MoveLeft**: Move to the left by a specified distance (meters).
  Format: `MoveLeft,distance [value, e.g., 0.3]`
- **MoveRight**: Move to the right by a specified distance (meters).
  Format: `MoveRight,distance [value, e.g., 0.3]`
- **RotateLeft**: Turn left by a specified number of degrees (0-180).
  Format: `RotateLeft,degrees [value, e.g., 30.2]`
- **RotateRight**: Turn right by a specified number of degrees (0-180).
  Format: `RotateRight,degrees [value, e.g., 30.2]`
- **LookUp**: Tilt your view up by a specified number of degrees (0-180).
  Format: `LookUp,degrees [value, e.g., 30.2]`
- **LookDown**: Tilt your view down by a specified number of degrees (0-180).
  Format: `LookDown,degrees [value, e.g., 30.2]`
- **Done**: Choose this action **only when you are less than 1.5 meter from the target, and the crosshair is precisely centered on the target object**.
  Format: `Done`

Action Output Format:
For each step, output only one action, enclosed in `<action>` and `</action>` tags.
Example:
`<action> RotateRight,degrees 25 </action>`
Interaction Process:

- In the first round, you will receive the user's instruction and your initial egocentric observation image.
- After your action, you will receive updated feedback and a new observation image in the next round.
- Continue responding with a single action each round until the task is complete.

EXPLORATION: System Prompt

You are an embodied robot working inside a room. Your task is to **actively explore the room to answer visual questions about objects and receptacles**.

Task Overview

You will be given a multiple-choice question about objects in the room. Your goal is to explore the environment, observe objects and their locations, and determine the correct answer.

Task Completion Process

1. **Exploration Phase**: Navigate through the room to gather visual information needed to answer the question.
2. **Answer Phase**: Once you have sufficient information, provide your final answer.

Available Actions

- **MoveAhead**: Move forward by a specified distance (meters).
  Format: `MoveAhead,distance [value, e.g., 0.3]`
- **MoveBack**: Move backward by a specified distance (meters).
  Format: `MoveBack,distance [value, e.g., 0.3]`
- **MoveLeft**: Move to the left by a specified distance (meters).
  Format: `MoveLeft,distance [value, e.g., 0.3]`
- **MoveRight**: Move to the right by a specified distance (meters).
  Format: `MoveRight,distance [value, e.g., 0.3]`
- **RotateLeft**: Turn left by a specified number of degrees (0-180).
  Format: `RotateLeft,degrees [value, e.g., 30.2]`
- **RotateRight**: Turn right by a specified number of degrees (0-180).
  Format: `RotateRight,degrees [value, e.g., 30.2]`
- **LookUp**: Tilt your view up by a specified number of degrees (0-180).
  Format: `LookUp,degrees [value, e.g., 30.2]`
- **LookDown**: Tilt your view down by a specified number of degrees (0-180).
  Format: `LookDown,degrees [value, e.g., 30.2]`

Output Format

During Exploration:

- Output only one action per round, enclosed in `<action>` and `</action>` tags.
- *Example:* `<action> RotateRight,degrees 25 </action>`

When Ready to Answer:

- Output only the option number (0, 1, 2, or 3) enclosed in `<answer>` and `</answer>` tags.
- *Example:* `<answer> 2 </answer>`

Interaction Flow

- You will receive a question with multiple choice options and an initial egocentric observation image.
- Explore the room using the available actions to get a comprehensive view and gather sufficient information.
- After each action, the environment feedback and updated observation image will provided.
- Continue exploring until you can confidently answer the question.
- Provide your final answer using the specified format.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The Experiment Section

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

Justification: The Conclusion Section

Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The Experiment Section

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: In the Experiment Section

   Guidelines: The Evaluation Setup subsection of the Experiment Section, together with the Appendix

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

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

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

   Answer: [Yes]

   Justification: We provided the anonymous repo link in the Appendix

   Guidelines:

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

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The Experiment section

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The Experiment section

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: In the Appendix

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: In the Experiment Section and Appendix

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: In the Conclusion section

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [Yes]

    Justification: In the Appendix

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: In the Appendix

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: In the Appendix

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: We don't use crowdsourcing or research with human subjects.

Guidelines:

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

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

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

    Answer: [NA]

    Justification: We don't use crowdsourcing or research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: In the Experiment section and the Appendix

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.