Embodied-Reasoner: Synergizing Visual Search, Reasoning, and Action for Embodied Interactive Tasks

Wenqi Zhang[1,*], Mengna Wang[2,3,*], Gangao Liu[2,3], Xu Huixin[2,6,7], Yiwei Jiang[2,6,8], Yongliang Shen[1], Guiyang Hou[1], Zhe Zheng[1], Hang Zhang[4], Xin Li[5], Jiajun Liu[9], Weiming Lu[1,†], Peng Li[2,3,6,†], Yueting Zhuang[1,†]

[1]Zhejiang University, [2]Institute of Software, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences [4]Alibaba Group
[5]DAMO Academy, Alibaba Group [6]Nanjing Institute of Software Technology
[7]Nanjing University of Posts and Telecommunications
[8]Hohai University [9]Renmin University of China
zhangwenqi@zju.edu.cn, lipeng@iscas.ac.cn

Abstract

Recent advances in reasoning models have demonstrated remarkable capabilities on mathematical and coding tasks. However, their effectiveness in embodied domains, where the agent must continuously interact with environments and process observation-action interleaved trajectories, remains largely unexplored. We present *Embodied-Reasoner*, a reasoning model for interactive embodied tasks. Unlike mathematical reasoning that relies primarily on logical deduction, embodied scenarios demand spatial understanding, temporal reasoning, and ongoing self-reflection based on interaction history. To address these challenges, we synthesize 9.3k coherent Observation-Thought-Action trajectories containing 64k ego-centric images and 90k diverse reasoning processes (analysis, spatial reasoning, reflection, planning, and verification). We develop a three-stage training recipe that progressively enhances the model's capabilities through imitation learning, rejection sampling tuning on self-exploration trajectories, and reflection tuning. The evaluation shows that our model significantly outperforms advanced visual reasoning models, e.g., exceeds OpenAI o1, o3-mini, and Claude-3.7 by +9%, 24%, and +13%. Analysis reveals that our model exhibits fewer repeated searches and logical inconsistencies, with particular advantages in complex long-horizon tasks. Testing on unseen scenarios and real-world also validates our generalization [1].

1 Introduction

Recent advancements in reasoning models such as OpenAI o1 (OpenAI, 2024c), DeepSeek R1 (Guo et al., 2025), and Qwen-QwQ (Team, 2024) have demonstrated significant progress in domains requiring extensive deliberation, e.g., college-level mathematics (Min et al., 2024) and coding tasks (Zhang et al., 2024g; Huang et al., 2025b). These models, trained through reinforcement learning with verifiable rewards (RLVR) (Guo et al., 2025) or post-training on elaborate thinking trajectories (Zhao et al., 2024a; Min et al., 2024), exhibit human-like reasoning and reflection.

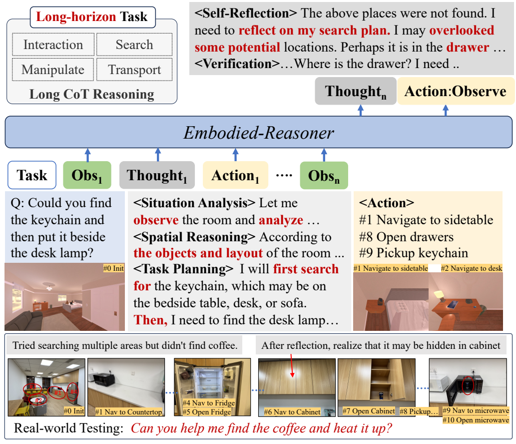


Figure 1: Our embodied reasoning model can actively **observe** and **reason**,**plan** detailed search paths, **interact** with objects (e.g., opening drawers, pickup) for inspection, and **reflect** on failed behaviors, achieving search, manipulation, and transportation of hidden objects.

Despite these advances, a critical question emerges: *Beyond these specialized domains, can deep thinking paradigm be extended to more general scenarios?* Particularly, applied to complex embodied tasks demanding long-horizon planning and deliberate reasoning in interactive environments (Shridhar et al., 2020)? This extension is non-trivial due to several fundamental challenges:

Challenge 1: Multi-turn interaction with environment brings **multimodal interleaved context**. Compared to many QA tasks, which are limited

*The first two authors have equal contributions.
†Corresponding author.
[1]Code is in https://github.com/zwq2018/embodied_reasoner

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 41178–41207
July 2-7, 2026

Figure 2: *Embodied-Reasoner* exhibits spontaneous thinking behaviors, e.g., *analyzing environmental states* (#1,3), *reflecting on missed details* (#4), *reasoning based on the latest observations* (#5), and *recalling cues for efficient planning* (#9). These thoughts remain coherent and logically consistent despite spanning multiple rounds.

to **single-turn** dialogues, embodied agents operate in an **interactive manner** over long-horizon task: continuously interact with the environment, gather real-time observation (mostly visual modality), then carefully reason and act (textual modality). Therefore, agent needs to process **lengthy** and image-text interleaved context and perform reasonable reasoning. Nevertheless, it is challenging for many multimodal models (Du et al., 2025; Guo et al., 2024; Yao et al., 2024). We observe that even advanced models like o1 or Gemini 2.5 (OpenAI, 2025; DeepMind, 2025) frequently fail to exhibit robust reasoning in these interactive tasks, leading to repetitive or inconsistent behaviors.

Challenge 2: Embodied tasks require **diverse reasoning abilities**. Unlike mathematical tasks that mostly rely on **knowledge** and **logical deduction**, embodied scenarios demand a broader set of capabilities grounded in daily life. As shown in Figure 2, when searching for an object hidden in an unknown room, the agent must leverage **commonsense** to infer potential locations (*steps 1, 3*), explore unknown areas using **spatial reasoning** (*steps 1, 5*), and **recall** key clues from previous explorations (*step 9*) while **reflecting** on prior failures. These multifaceted abilities pose significant challenges for current models.

In this paper, we present *Embodied-Reasoner*, an end-to-end design that extends deep-thinking capabilities to embodied interactive models. Our insight is embodied tasks require ① comprehensive understanding of multimodal interaction, and also ② need different thinking patterns (spatial analysis, temporal recall, action reflection) for various situations or unexpected cases, e.g., planning exploration paths or responding to previous failures.

To this end, we develop an embodied data engine that automatically synthesizes *Observation-Thought-Action* trajectories with diverse environment interactions and embodied-specific reasoning processes. These coherent, image-text interleaved trajectories enable the model to develop robust spatial and temporal reasoning capabilities through learning from rich interaction experiences. We further design a three-stage training pipeline: Imitation→Exploration→Self-correction, which progressively equips the model with basic interaction skills, enhanced exploration abilities, and robust self-correction capabilities for handling failures and unexpected situations.

We evaluate our model on four high-level embodied tasks in AI2-THOR (Kolve et al., 2017; Shridhar et al., 2020): search, manipulation, transportation, and composite task. These tasks require the agent to search for hidden objects in unfamiliar room by leveraging commonsense knowledge to infer potential locations, spatial reasoning to plan efficient exploration paths, and actively interacting with containers (e.g., opening fridges, drawers) for inspection. The model must also reflect on failed behaviors and adapt its plan accordingly.

Evaluating on 809 novel tasks, *Embodied-Reasoner* significantly outperforms advanced reasoning VLMs, e.g., OpenAI o1, o3-mini, and

Claude-3.7 by +9, +24 and +22%, with more pronounced gains on complex composite tasks (+39.9%). It exhibits more intelligent behavior through spatial reasoning and self-reflection, avoiding repetitive or unreasonable actions.

2 Observation-Thought-Action Corpora

To develop reasoning models in embodied scenarios, we design an interactive task requiring high-level planning, reasoning and object interaction rather than low-level motor control, i.e., search for hidden objects (Sec. 2.1). We then introduce an embodied data engine that synthesizes task instructions (Sec. 2.2) and interaction trajectories, including action sequences (Sec. 2.3) and step-level reasoning (Sec. 2.4). Each trajectory includes multi-turn interactions, creating an interleaved context: obs → thought →action→obs→thought...

Figure 2 shows various cases adopting different thinking patterns: spatial reasoning for room layout and object relations, commonsense reasoning to search potential areas, reflection on plans.

2.1 Embodied Interactive Task

Task Environments. We built our task environment using AI2-THOR simulator with 120 indoor scenes (e.g., kitchens, bathrooms) and 2,100 objects (e.g., credit card, microwave). We control robot movements (e.g., moveahead) and interactions (e.g., pickup, open) via AI2THOR's API, capturing visual observations at each step.

Task Categories. The robot initializes in a corner of an unknown room with limited view. We design four common tasks with different complexities: ① **Search:** Finding an object in an unknown room, e.g., keychain, possibly hidden inside a container. ② **Manipulate:** Interacting with objects after searching, e.g., "*finding a lamp and turning on the switch*". ③ **Transport:** Finding a hidden object and transporting it to another location, involving multiple search and manipulation steps. ④**Composite task:** Multiple transport tasks in sequence, e.g., *"Place the egg in the microwave, then put it on the desk after heating..."*.

Action Definition. Since our task focuses on high-level planning rather than movement control, and low-level actions may cause excessive interactions, we encapsulate 9 high-level actions: *Observe*, *Move Forward*, *Navigate to {}*, *Put in {}*, *Pickup {}*, *Toggle {}*, *Close {}*, *Open {}*, *Termination*.

2.2 Scene-Constrained Instruction Synthesis

We employ LLMs to synthesize task instructions automatically. Unlike most LLM-based synthesis methods, embodied instructions must satisfy physical scenario constraints—instructions cannot refer to non-existent objects or involve illegal actions. E.g., "*Please move sofa to corner*" is invalid if the room lacks a sofa or sofa cannot be moved. Thus, we design task templates, leverage GPT-4o's coding capabilities to select objects meeting constraints, then diversify instructions into different styles and complexities.

Task Templates with Constraints. We design multiple templates for each task. Figure 3 presents a transport task template: *pickup {invisible A} and put in {B}*, where A denotes a hidden pickupable object, e.g., keychain, and *B* should be a container like drawer or desk. This ensures instruction validity. See Tables B8 and B9 for details.

Code-based Object Filter. GPT-4o selects an appropriate template and generates code for constraint checking based on metadata, selecting objects satisfying constraints. We fill templates with matched objects (*A*: keychain, *B*: desk), synthesizing different instructions with many combinations.

Diversify Instructions. Lastly, we diversify instructions at two levels: ① Style: GPT-4o rewrites filled templates into human-style instructions, e.g., "*I can not find my keychain. Can you help me find them and ...*". ② Difficulty: We combine multiple simple tasks to create composite tasks.

2.3 Action Sequence Synthesis

We first annotates the key actions sequences for each instructions and then add diverse search processes into key actions.

Affiliation Graph. As shown in Figure 3, we construct an affiliation graph using simulator metadata. Each node represents an object, and edges denote affiliation relations, e.g., keychain in a drawer is depicted as leaf (keychain) connected to parent node (drawer) with an "include" relationship.

Key Action Sequence. We utilize the affiliation graph to derive minimum required action sequence (key actions) for task completion. For "*pickup keychain and place it in desk*", we trace from leaf node (keychain) upward to parent (drawer) and finally to desk. GPT-4o generates corresponding actions: *A1: Navigate to Mudroom*, *A2: Navigate to Drawer*, *A3: Open Drawer*, *A4: Pickup ...*. All key actions are indispensable for task completion.

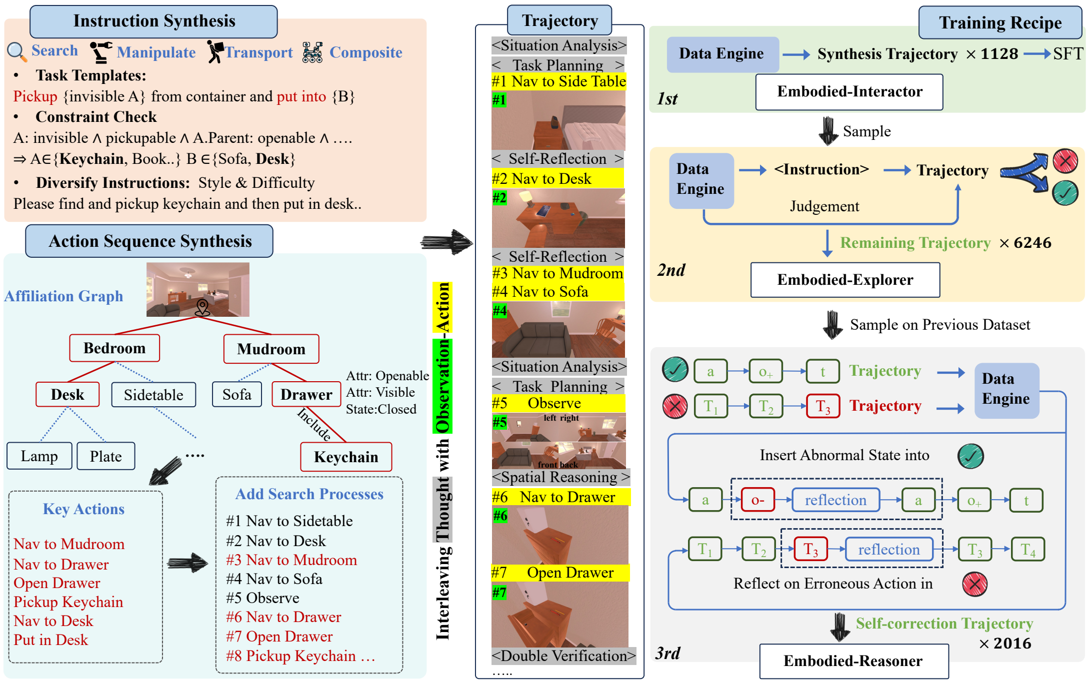


Figure 3: *Left*: We first synthesize instructions from task templates, build an affiliation graph to derive key actions, add exploratory actions, and insert thinking thoughts between observations and actions. *Right*: Three-stage training recipe. ① Fine-tune on synthesized trajectories to develop interaction skills. ② Sample multiple trajectories on novel tasks and use successful ones to develop exploring abilities. ③ Inject anomalous states and reflective thoughts in successful cases while correcting errors in failed ones for training *Embodied-Reasoner*.

Add Additional Searching Processes. Beyond key action sequences, our engine synthesizes exploratory paths by inserting additional search processes. As shown in Figure 3, our engine inserts three searching actions: *Nav to Sidetable*, *Desk*, and *Sofa*. After failing to find the keychain, it inserts an *Observe* action until locating the keychain in the drawer. These additional search actions make trajectories more realistic, showcasing how robots gradually explore unfamiliar environments until it successfully locates the target.

2.4 Interleave Thought with Obs-Action

After running synthesized actions ($a_{1:n}$), we obtain trajectory: $o_1, a_1, ..., o_n, a_n$. We then generate reasoning thoughts (t_i) for each step, forming interleaved context: *Obs-Thought-Action*.

Diverse Thinking Pattern. We define five thinking patterns simulating human cognitive activities: *Situation Analysis, Task Planning, Spatial Reasoning, Self-Reflection, and Self-Verification*. We describe each pattern in detail, guiding GPT-4o in synthesizing corresponding thinking processes.

Derive Thought from Observation-Action. For each interaction, GPT-4o selects reasonable thinking patterns and generates detailed thoughts based on interactive context, inserted between observations and actions (o_n,a_n → o_n,t_n,a_n). Specifically, prompted by previous trajectory $(..., o_{t\text{-}1}, t_{n\text{-}1}, a_{n\text{-}1}, o_n)$ and upcoming action (a_n), GPT-4o generates a well-reasoned thinking process (t_n) to explain a_n. It may contain multiple thinking patterns (e.g., analysis, planning) while remaining logically consistent with previous thoughts ($t_{1:n\text{-}1}$).

3 Training Recipe

To incentivize reasoning ability, we design three training stages: imitation learning, rejection sampling tuning, and reflection tuning, bootstrapping a general VLM into an embodied interactive model with deep thinking ability. For training this image-text interleaved context, each trajectory is formatted as **multi-turn dialogue corpus** with loss computed only for reasoning and action tokens.

3.1 Learn to Interact: Imitation Learning

Firstly, we synthesizes a small set of simple <task instruction, trajectory>. Most tasks are relatively simple without complex search and reasoning. We fine-tune Qwen2-VL-7B-Instruct on these trajectories, developing ***Embodied-Interactor***.

After training, *Embodied-Interactor* acquires basic interaction abilities: understanding interleaved image-text input and generating reasoning and action tokens. However, since most trajectories lack complex reasoning or exploration, *Embodied-Interactor* exhibits limited visual search capabilities. When target objects require further searching, it may hallucinate or give up. E.g., it opens a fridge for an egg but finds it empty, then responds "egg does not exist" rather than searching elsewhere.

3.2 Learn to Search: Rejection Sampling

Self-generated Trajectory for Novel Task. To incentivize exploration ability, we apply rejection sampling tuning on self-generated trajectories. As shown in Figure 3, data engine synthesizes new tasks and their key actions. For each task, *Embodied-Interactor* samples multiple trajectories under high temperature, some succeeding after multiple visual searches.

Rejection Sampling by Data Engine. For each self-generated trajectory, our data engine evaluates task completion, i.e., whether it contains all key actions in proper order and achieves the correct final state. After rejecting failed trajectories, we collect 6,246 successful samples for instruction tuning, developing *Embodied-Explorer*. It exhibits adaptive planning and searching behaviors—when targets cannot be found directly, it formulates search plans involving many potential areas with priorities.

3.3 Learn to Self-reflect: Reflection Tuning

Embodied-Explorer occasionally produces unreasonable actions in long-horizon tasks, e.g., hallucination. Besides, robots often encounter temporary hardware malfunctions. This requires the model to self-reflect on unreasonable behaviors, recognize abnormal states, and correct them timely. As shown in Figure 3, *Embodied-Explorer* samples massive trajectories on previous tasks. ① For succeeded trajectories, we insert anomaly states to simulate hardware fault. ② For failed trajectories, we locate the first erroneous action and construct self-correction trajectories.

Insert Abnormal State into Succeeded Trajectory. We simulate two robot anomalies: **navigation anomaly**, where the robot navigates to an incorrect location (e.g., "*navigate to fridge*" but arrives at table); and **manipulation anomaly**, where robot arm temporarily fails to execute commands. For a succeeded trajectory $\{.., a, o_+, t..\}$, we insert an abnormal state (o_-) after action (a), generate self-reflective thoughts (t_r):$\{.., a, o_-, t_r, a, o_+ ..\}$.

Stage	#Trajectory	Source	#Img$_{all}$	#Img$_{max}$	#Action$_{avg}$
Train$_{1st}$	1,128	Synthesis	4636	11	4.11
Train$_{2nd}$	6,246	Self-Explore	45.8k	26	7.33
Train$_{3rd}$	2,016	Synthesis	13.8k	29	8.63
Total	9,390	-	64k	29	7.22
Testset	809	Human	4.9k	29	6.06

Table 1: We synthesize 9.3k ⟨task, trajectory⟩ for training and annotate key actions for 809 novel testing tasks.

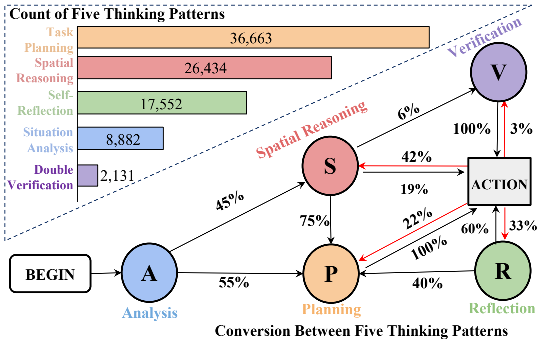


Figure 4: We analyze the frequency of five types of thoughts and their flexible transition relationships.

Reflect on Unreasonable Action in Failed Trajectory. Using ground-true actions, we identify the first incorrect action in each failed trajectory ($Traj_-$). We generate self-reflective thoughts (t_r) and supplement remaining correct trajectories ($Traj_+^{t:n}$), creating: $\{Traj_-^{1:t}, t_r^t, Traj_+^{t:n}\}$. We finetune on synthesized self-correction trajectories, masking erroneous partial trajectory and computing loss only for reflective tokens and correct trajectory.

4 Dataset Statistics

We synthesize 9,390 ⟨*Scene, Instruction, Trajectory*⟩ across 3 stages, with each trajectory following *Obs-Thought-Action*. As shown in Table 1, data engine synthesizes 1,128 samples in the first stage. In the 2nd stage, we remain 6,246 exploratory trajectories via rejection sampling. The 3rd stage synthesizes 2,016 self-correction trajectories.

Dataset Distribution. The distribution is presented in Section B. It covers 107 indoor scenes (e.g., kitchens, living rooms), 2,100 objects (e.g., eggs, laptops) and 2,600 containers (e.g., fridge, drawers), containing *64k* ego-centric images from interaction and *8M* reasoning tokens.

4.1 Thoughts Analysis

The Distribution of Thinking Patterns. We count the frequency of five thinking patterns in all trajectories. As shown in Figure 4, *task planning* and *spatial reasoning* appear most frequently, with

Model		Success Rate ↑	Search Efficiency ↑	Task Completeness ↑	Success Rate for SubTasks ↑			
					Search	Manipulate	Transport	Composite
General-purpose VLMs	Qwen2.5-VL-72B-Instruct (2025)	31.75%	22.61%	50.62%	52.14%	38.89%	21.90%	0.00%
	Qwen2-VL-72B-Instruct (2024b)	39.00%	28.88%	54.56%	50.00%	52.36%	33.19%	0.00%
	Claude 3.5-Sonnet (2024)	45.35%	28.05%	64.12%	54.25%	50.51%	51.22%	3.84%
	Qwen-VL-Max (2025b)	49.81%	36.28%	68.39%	63.87%	63.21%	45.16%	1.90%
	Llama-4-Scout-17B-16E (2025)	51.68%	37.42%	71.23%	69.73%	70.44%	40.34%	14.28%
	GPT-4o (2024b)	66.67%	41.68%	79.07%	69.03%	79.26%	71.95%	14.42%
Visual Reasoning Models	QVQ-72B-Preview (2025)	7.54%	6.39%	36.33%	4.35%	7.50%	10.53%	0.00%
	Kimi-K1.5† (2025c)	46.00%	-	-	-	-	-	-
	GPT-o3-mini (2025)	56.55%	26.93%	67.41%	**78.57%**	59.32%	66.67%	0.00%
	Gemini-2.0 Flash Thinking (2025)	56.74%	43.01%	71.70%	71.05%	75.60%	40.67%	8.89%
	Claude-3.7-Sonnet-thinking (2025)	67.70%	37.95%	78.63%	69.12%	75.88%	71.94%	13.79%
	GPT-o1 (2024c)	71.73%	43.06%	82.49%	78.42%	79.10%	67.36%	13.16%
	Embodied-Interactor-7B (ours-1st)	25.46%	24.75%	53.67%	30.97%	27.09%	29.20%	3.81%
	Embodied-Explorer-7B (ours-2nd)	65.39%	46.25%	77.73%	60.00%	75.92%	72.24%	26.67%
	Embodied-Reasoner-7B (ours-3rd)	**80.96%**	**55.07%**	**86.30%**	65.16%	**93.31%**	**87.20%**	**54.29%**

Table 2: We compare *Embodied-Reasoner* against general and reasoning VLMs. After three-stage training, we boost Qwen2-VL-7B from 14.8 to 25.4→65.3→81.9. † means manually evaluate 50 testing cases thought webpage.

36.6k and 26.4k occurrences, averaging four and three times per trajectory respectively. Besides, we observe *Self-reflection* typically occurs after search failures, averaging twice per trajectory.

Transition between five Thinking Patterns. We compute the transition probabilities between five patterns by recording the frequency of each pair (e.g., A→S, A→P) and normalizing by the frequency of each state. In Figure 4, we notice several interesting phenomena. It typically begins with *situation analysis* (8882 times), followed by *planning* (4886 times, 55%) and *spatial reasoning* (3996 times, 45%). When navigating unknown regions, it frequently relies on *spatial reasoning* (Action→S: 42%). If a search attempt fails, it shifts to *self-reflection* (Action→R: 33%), and once a sub-task is completed, it may perform *verification* sometimes (Action→V: 3%). This flexible transformation enhances the model's autonomy and adaptability.

4.2 Interactive Evaluation Framework

We cultivate 809 test tasks across 12 novel scenarios (different from training). We manually annotate the key actions and final states for each task: ⟨task, Key Action, Final state⟩. Besides, it contains 25 carefully designed ultra long-horizon tasks, each involving four subtasks and 14-to-27 key actions.

Metrics. Three metrics to assess trajectory quality. **Success Rate (SR)**: Whether contains all key actions in order and the final state is also correct. **Search Efficiency**: Ratio of key actions to predicted sequences ($N_{ak} \div N_{pt}$). **Task Completeness**: Proportion of completed key actions ($N_{pk} \div N_{ak}$), where N_{pk} denotes predicted key actions, N_{pt} denotes total predicted actions, and N_{ak} denotes annotated key actions.

5 Experiments

5.1 Main Results

Higher success rate, task efficiency, and completeness. As shown in Table 2, *Embodied-Reasoner* significantly outperforms all general and reasoning VLMs (+9.6% over GPT-o1), including GPT o3-mini (+24%) and Claude-3.7-Sonnet-thinking (+13%). It also demonstrates clear advantages on search efficiency and task completeness, e.g., +12% higher than GPT-o1. It reflects our more efficient interaction process, completing multi-step searching and reasoning with fewer actions.

Advantages are more pronounced on complex tasks. Across four sub-task types, *Embodied-Reasoner* shows increasingly stronger advantages as task complexity grows. On simpler manipulation tasks, our model achieves performance comparable to baseline VLMs (93% vs. 79%). However, on challenging composite tasks requiring multi-step reasoning and exploration, it substantially outperforms the second-best model (GPT-4o) by +39.9%, highlighting its capability in handling complex, long-horizon scenarios.

Three-stage training progressively incentivizes reasoning capabilities, from 14.7% to 80.9%. Our base model, Qwen2-VL-7B, initially achieved only 14.7%. After first-stage imitation learning, it improved to 25.4%, mastering basic interaction ability. Rejection sampling tuning then significantly boosts performance to 65.4%, reaching GPT-o1 level. Finally, fine-tuning with self-correction trajectories further elevated to 80.9%. Through three-stage, our model learns to perform strategic search and spatial reasoning after initial planning, with timely self-reflection, reduc-

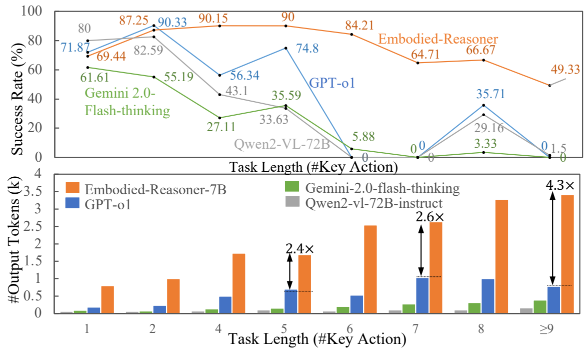


Figure 5: Relations between *Task Length-Success Rate*, and with *Output Token*. As task complexity increases (Length ↑), our model use more tokens for task-solving.

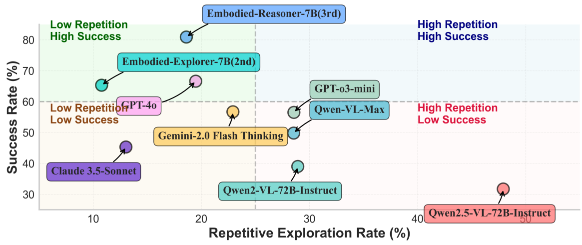


Figure 6: Relation between repetitive exploration rate (x-axis) and success rate (y-axis). Baselines often exhibit repetitive searching of the same areas, while our model achieves low repetition and high success after thinking.

ing repetitive searching and unreasonable planning—common errors often observed in baselines.

5.2 Performance across Task Difficulties

We denote *task length* as the number of key actions required for each task. Larger task length indicates greater complexity requiring more interactions.

Robustness on long-horizon tasks. As illustrated in Figure 5 (upper), we visualize the relationship between task length and success rate. As task length increases, baseline performance degrades substantially, with some models failing entirely when tasks exceed five interactions. In contrast, *Embodied-Reasoner* maintains robust performance, consistently achieving over 60% across most cases.

Employing longer CoT reasoning for complex tasks. Figure 5 (Bottom) shows *Embodied-Reasoner* allocates more reasoning tokens than other thinking LLMs, and this trend becomes more pronounced as task length increases. When task length is 5, *Embodied-Reasoner*'s chain of thought length is 2.4× that of Gemini-2.0-flash, and when task length ≥ 9, this ratio becomes 4.3×. Our model engages in significantly longer analysis and more deliberate self-reflection for uncertain situations. Although this requires longer reasoning time, it yields significant performance improvements.

5.3 Analysis on Model's Behaviors

Reasoning paradigm reduces repetitive searching behaviors. To quantify searching efficiency of a trajectory, we define a **repetitive exploration rate (RER)**, which measures how often the robot navigates to the same area within its trajectory. As shown in Figure 6, our models exhibit a significantly **lower RER** (18%) and **higher success** rate (80%). In contrast, most baseline models exhibit low success rates and high RER, e.g., GPT-o3-mini achieves only 56% success rate with a RER as high as 28%. Upon careful examination of their trajectories, we find due to weak temporal reasoning and context awareness, baselines frequently **revisit the same locations and perform repetitive searches**. However, our model is more intelligent: through memory retrospection and reflection, it reduces repetitive searching and improves success rates.

6 Generalization

6.1 Generalization to More Diverse Scenarios

We also assess on ① in-distribution scenarios with the same simulator (ALFRED), ② OOD scenarios from other simulators, and ③ real-world scenarios.

① **Testing on different benchmark (ALFRED).** As shown in Table 3, our model achieves a competitive 56% success rate, approaching Gemini-2.5-Pro-thinking (57.81%) and Robotics-ER 1.5 (60%), showing strong generalization for different benchmarks. Notably, we observe a clear complementarity between two paradigms: embodied model excels at handling long-horizon tasks with complex, ambiguous instructions through flexible planning and adaptation (act as planner), while IL/RL-based expert models demonstrate superior efficiency and precision on well-defined, specific tasks due to their specialized training (as low-level actuator). See Table A1 for more results.

② **Testing on OOD scenarios from other simulators**. To assess model's generalization capability across different visual domains and scene shifts, we evaluate on two benchmarks from different simulators: ① R2R (Anderson et al., 2018) benchmark on Habitat simulator (Savva et al., 2019) with Matterport3D scenes (Chang et al., 2017), and ② EmbodiedBench(Yang et al., 2025a) on Habitat with YCB (Calli et al., 2015) and ReplicaCAD (Szot et al., 2021). As in Figure 7 and tables A5 and A6,

Types	Models	SR(%)
Imitation Learning or RL-based Expert Models	MOCA(Singh et al., 2021)	5.36
	HiTUT(Zhang and Chai, 2021)	10.23
	HLSM(Blukis et al., 2022)	18.28
	FILM(Min et al., 2021)	20.10
	LGS-RPA(Murray and Cakmak, 2022)	30.41
Embodied Agent	LLM-Planner(Song et al., 2023)+4o-mini	22.54
General VLMs	Qwen2.5-VL-72B-Instruct	35.49
	GPT-4o-mini (2024a)	37.72
	Gemini-2.5-Pro-thinking (2025)	57.81
Embodied Model	Gemini Robotics-ER 1.5(2025a)	60.71
	Embodied-Reasoner-7B	56.03

Table 3: Comparison with Imitation-, RL- and LLM-based agents on different benchmark (ALFRED).

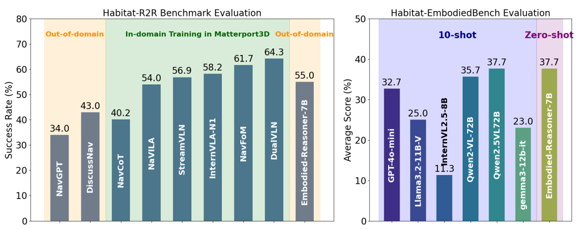


Figure 7: OOD evaluation on Habitat R2R and EmbodiedBench. Our model was not trained in these scenarios, yet it achieves comparable performance with these advanced VLN models (in-domain training).

our model achieves competitive performance (55% and 37.7%) against advanced VLN models, showing strong cross-simulator generalization despite no training on these visual domains or scenes.

③ **Testing in real-world Scenarios**. We evaluate *Embodied-Reasoner* on 30 real-world object searching tasks across 6 kitchen, 12 bathroom, and 12 bedroom scenes. A human operator captures real-time observations with a handheld camera. Our model analyzes the image and generates an action command, which the operator manually executes. Figure 8 shows an example: *"Can you help me find the coffee and heat it up?"* Our model rules out the countertop and dining table after two explorations (steps 1,2), ultimately locating the coffee (#7) in the cabinet and placing it in the microwave for heating (#11). In contrast, OpenAI o3-mini ① **fails to formulate a reasonable plan**, ② **forgets the assigned task**, and ③ **search the same area repeatedly 3 times**. See Table A7 for details.

6.2 Low-Latency with Context Memory

Embodied-Reasoner also can be seamlessly augmented with a context memory mechanism to handle long-horizon tasks more efficiently. The memory mechanism works as follows: historical trajectories (interleaved image-text context) are compressed into concise textual summaries paired with image indices, while all historical images are stored

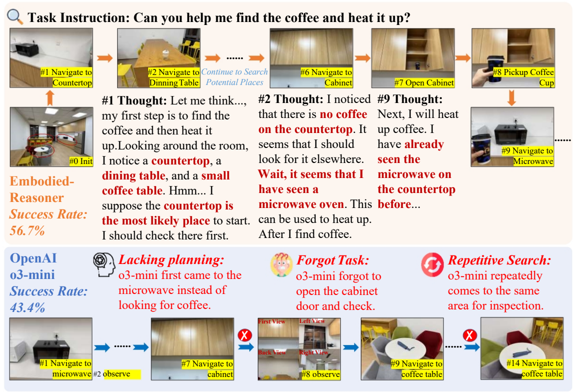


Figure 8: Real scenarios testing. We achieve a higher success rate (56.7%) than o3-mini and o1 (43.4%, 50%).

in a retrieval buffer. During interaction, the model receives the task instruction, trajectory summary, and current observation. It then either ① directly outputs actions based on current observation, or ② actively retrieve relevant images from image buffer by issuing a special action `image-view {index}`, enabling selective access to historical information without overwhelming the context.

To achieve this, we manually craft 50 memory-augmented trajectories and integrate them into the original dataset. An example is provided in Figure B13. The results of memory-augmented model are shown in Table A2: consumed token are dramatically reduced by 90% (from *52k* to *5.8k* per trajectory) due to compressed context length, while success rate remains stable, showing efficient context management ability for long-horizon tasks.

7 Analysis on Training Recipe

7.1 Are Five Thinking Patterns Necessary?

As shown in Section 2.4, we design five thinking patterns: *Spatial Reasoning*, *Planning*, *Situation Analysis*, *Reflection*, and *Verification*. To investigate whether explicitly distinguishing these patterns is necessary, we ablate them by unifying all patterns under a single `<thinking>` tag. For example, `<reflection>`*"I found that my previous action was wrong..."*`</reflection>` `<planning>`*"I re-observe the room..."*`</planning>` are merged into `<thinking>`*"The previous plan has issues, I re-observe the room..."*`</thinking>`.

As shown in Table 4, training with a single pattern leads to performance degradation, with particularly notable drops in Transport (-4.40%) and Composite (-6.77%) tasks that require more complex reasoning. It shows that explicitly distinguishing

	Transport SR(%)	Composite SR(%)
w/ different thinking pattern	87.20	54.29
w/o different thinking pattern	82.80 (↓4.40)	47.52 (↓6.77)

Table 4: Ablation experiments on thinking patterns using only a single <thinking> tag, which lead to significant success rate drops on more challenging Transport and Composite tasks.

Stages			Success	Search	Task	Success Rate for SubTasks	
1st	*2nd*	*3rd*	Rate	Efficiency	Completeness	Transport	Composite
✓	✓	✓	**81.0%**	**55.1%**	**86.3%**	**87.2%**	**54.3%**
✗	✓	✓	19.1%	11.6%	43.6%	24.2%	0.9%
✓	✗	✓	52.1%	27.8%	65.1%	61.3%	9.4%
✓	✓	✗	65.4%	46.3%	77.7%	72.2%	26.7%

Table 5: Ablation experiment on three training stages (*Stage1 Imitation Learning*, *Stage2 Rejection Sampling Tuning*, *Stage 3 Reflection Tuning*).

thinking patterns helps regulate the model's reasoning process by providing structured cognitive scaffolding, e.g., *Reasoning* guides location understanding, *Planning* organizes action sequences. See Table A3 for details.

7.2 Impact of Three Training Stages

We systematically ablate each training stage. As shown in Table 5, removing Stage 1 (Imitation Learning) causes a drastic performance drop (from 81.0% to 19.1%), as the model fails to collect sufficient samples for rejection sampling. It indicates that Stage 1 provides crucial warm-up for learning the thinking format and interaction behaviors. Removing Stage 2 (Rejection Sampling Tuning) also leads to substantial degradation (to 52.1%), primarily due to insufficient environment exploration. Stage 3 mainly enhances the model's ability to recover from errors and reflect, as evidenced by the significant performance improvement on Composite tasks (from 26.7% to 54.3%). See Table A4 for details.

8 Discussions

We discuss two important considerations in our data engine and training pipeline: addressing the bias toward easy tasks in the data distribution, and guiding model relies on active visual perception rather than oracle simulator metadata. Specifically:

- In Stage 2 (Rejection Sampling Fine-Tuning), we adopt an iterative multi-round strategy to progressively unlock the model's capabilities on complex tasks. For failed attempts, we incorporate hints to guide the sampling of hard scenarios. Furthermore, in Stage 3, the failed trajectories are utilized to construct self-correction data, which guides the model to self-reflect on unreasonable behaviors, and to recognize and correct abnormal states.

- First, while the tasks and key action sequences are derived from simulator metadata, our reasoning thoughts are synthesized strictly based on actual visual observations at each step. Second, we inject plausible exploratory actions into the key action sequences, which guides the model to learn visual search process instead of memorizing object locations.

9 Conclusions

We propose an embodied reasoning model for interactive tasks that can spontaneously search, reason, and act. To develop this, we design a data engine that synthesizes 9,390 interactive trajectories in an *Observation-Thought-Action* interleaved format. We employ a three-stage training pipeline—to progressively enhance its interaction and reasoning abilities. Extensive evaluations demonstrate that our model exhibits superior reasoning capabilities.

Limitations

Our work has two main limitations. First, although our model demonstrates promising generalization and transferability in OOD scenarios and also real-world experiments, deploying embodied agents in more complex and dynamic real-world environments remains challenging. Future work should explore more diverse real-world scenarios to further validate and improve robustness. Second, our current approach relies on discrete high-level actions and does not directly output low-level motor commands. Integrating our reasoning model with continuous control policies for more fine-grained manipulation tasks is an important direction for future work.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62436007), Science and Technology Major Project of Jiangsu Province (No.BG2024041) and Nanjing Science and Technology Plan (No. Y23002ZX01).

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, and 1 others. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. 2024. Etpnav: Evolving topological planning for vision-language navigation in continuous environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683.

Anthropic. 2024. Claude 3.5-sonnet: Anthropic's advanced language model.

Anthropic. 2025. Claude 3.7 sonnet: Anthropic's hybrid reasoning ai model.

Alisson Azzolini, Junjie Bai, Hannah Brandon, Jiaxin Cao, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, and 1 others. 2025. Cosmos-reason1: From physical common sense to embodied reasoning. *arXiv preprint arXiv:2503.15558*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, and 1 others. 2024. pi-0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*.

Valts Blukis, Chris Paxton, Dieter Fox, Animesh Garg, and Yoav Artzi. 2022. A persistent spatial semantic representation for high-level natural language instruction execution. In *Conference on Robot Learning*, pages 706–717. PMLR.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, and 1 others. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*.

Shaofei Cai, Zihao Wang, Kewei Lian, Zhancun Mu, Xiaojian Ma, Anji Liu, and Yitao Liang. 2024. Rocket-1: Mastering open-world interaction with visual-temporal context prompting. *arXiv preprint arXiv:2410.17856*.

Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. 2015. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3d: Learning from rgb-d data in indoor environments. *Preprint*, arXiv:1709.06158.

Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee Wong. 2024. Mapgpt: Map-guided prompting with adaptive path planning for vision-and-language navigation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9796–9810.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*.

Kevin Chen, Junshen K Chen, Jo Chuang, Marynel Vázquez, and Silvio Savarese. 2021a. Topological planning with transformers for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11276–11286.

Peihao Chen, Dongyu Ji, Kunyang Lin, Runhao Zeng, Thomas Li, Mingkui Tan, and Chuang Gan. 2022a. Weakly-supervised multi-granularity map learning for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 35:38149–38161.

Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. 2021b. History aware multimodal transformer for vision-and-language navigation. *Advances in neural information processing systems*, 34:5834–5847.

Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. 2022b. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16537–16547.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yiming Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, and 23 others. 2025. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *Preprint*, arXiv:2412.05271.

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. 2024. Navila: Legged robot vision-language-action model for navigation. *arXiv preprint arXiv:2412.04453*.

Reinis Cimurs, Il Hong Suh, and Jin Han Lee. 2021. Goal-driven autonomous exploration through deep reinforcement learning. *IEEE Robotics and Automation Letters*, 7(2):730–737.

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

InternNav Contributors. 2025. InternNav: InternRobotics' open platform for building generalized navigation foundation models. https://github.com/InternRobotics/InternNav.

Ishita Dasgupta, Christine Kaeser-Chen, Kenneth Marino, Arun Ahuja, Sheila Babayan, Felix Hill, and Rob Fergus. 2023. Collaborating with language models for embodied reasoning. *arXiv preprint arXiv:2302.00763*.

DeepMind. 2025. Flash thinking.

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. 2022. Procthor: Large-scale embodied ai using procedural generation. *Advances in Neural Information Processing Systems*, 35:5982–5994.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, and 1 others. 2023. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*.

Yifan Du, Zikang Liu, Yifan Li, Wayne Xin Zhao, Yuqi Huo, Bingning Wang, Weipeng Chen, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. 2025. Virgo: A preliminary exploration on reproducing o1-like mllm. *arXiv preprint arXiv:2501.01904*.

Kiana Ehsani, Tanmay Gupta, Rose Hendrix, Jordi Salvador, Luca Weihs, Kuo-Hao Zeng, Kunal Pratap Singh, Yejin Kim, Winson Han, Alvaro Herrasti, and 1 others. 2024. Spoc: Imitating shortest paths in simulation enables effective navigation and manipulation in the real world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16238–16250.

Kiana Ehsani, Winson Han, Alvaro Herrasti, Eli VanderBilt, Luca Weihs, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. 2021. Manipulathor: A framework for visual object manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4497–4506.

Georgios Georgakis, Karl Schmeckpeper, Karan Wanchoo, Soham Dan, Eleni Miltsakaki, Dan Roth, and Kostas Daniilidis. 2022. Cross-modal map learning for vision and language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15460–15470.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Jarvis Guo, Tuney Zheng, Yuelin Bai, Bo Li, Yubo Wang, King Zhu, Yizhi Li, Graham Neubig, Wenhu Chen, and Xiang Yue. 2024. Mammoth-vl: Eliciting multimodal reasoning with instruction tuning at scale. *arXiv preprint arXiv:2412.05237*.

Yicong Hong, Zun Wang, Qi Wu, and Stephen Gould. 2022. Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15439–15449.

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. 2022a. Visual language maps for robot navigation. *arXiv preprint arXiv:2210.05714*.

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. 2023a. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10608–10615. IEEE.

Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. 2023b. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022b. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, and 1 others. 2022c. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*.

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

Zhongzhen Huang, Gui Geng, Shengyi Hua, Zhen Huang, Haoyang Zou, Shaoting Zhang, Pengfei Liu, and Xiaofan Zhang. 2025b. O1 replication journey–part 3: Inference-time scaling for medical reasoning. *arXiv preprint arXiv:2501.06458*.

Shyam Sundar Kannan, Vishnunandan LN Venkatesh, and Byung-Cheol Min. 2024. Smart-llm: Smart multi-agent robot task planning using large language models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12140–12147. IEEE.

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*, pages 16373–16383.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, and 1 others. 2024. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, and 1 others. 2017. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. 2020. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pages 104–120. Springer.

Jonáš Kulhánek, Erik Derner, Tim De Bruin, and Robert Babuška. 2019. Vision-based navigation using deep reinforcement learning. In *2019 european conference on mobile robots (ECMR)*, pages 1–8. IEEE.

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

Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li E Li, Ruohan Zhang, and 1 others. 2024. Embodied agent interface: Benchmarking llms for embodied decision making. *Advances in Neural Information Processing Systems*, 37:100428–100534.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. 2023. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE.

Bingqian Lin, Yunshuang Nie, Ziming Wei, Jiaqi Chen, Shikui Ma, Jianhua Han, Hang Xu, Xiaojun Chang, and Xiaodan Liang. 2025. Navcot: Boosting llm-based vision-and-language navigation via learning disentangled reasoning. *Preprint*, arXiv:2403.07376.

Kevin Lin, Christopher Agia, Toki Migimatsu, Marco Pavone, and Jeannette Bohg. 2023. Text2motion: From natural language instructions to feasible plans. *Autonomous Robots*, 47(8):1345–1365.

Yuxing Long, Xiaoqi Li, Wenzhe Cai, and Hao Dong. 2023. Discuss before moving: Visual language navigation via multi-expert discussions. *Preprint*, arXiv:2309.11382.

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

Meta. 2025. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation.

Meta AI. 2024. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. Meta Blog. Accessed: 2026-01-06.

So Yeon Min, Devendra Singh Chaplot, Pradeep Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov. 2021. Film: Following instructions in language with modular methods. *arXiv preprint arXiv:2110.07342*.

Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, and 1 others. 2024. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems. *arXiv preprint arXiv:2412.09413*.

Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. 2023. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *Advances in Neural Information Processing Systems*, 36:25081–25094.

Michael Murray and Maya Cakmak. 2022. Following natural language instructions for household tasks with landmark guided search and reinforced pose adjustment. *IEEE Robotics and Automation Letters*, 7(3):6870–6877.

OpenAI. 2024a. Gpt-4o mini: advancing cost-efficient intelligence.

OpenAI. 2024b. Gpt-4o: Openai's multimodal language model.

OpenAI. 2024c. Openai o1: Advanced reasoning ai model.

OpenAI. 2025. o3-mini: Openai's advanced reasoning model.

Bowen Pan, Rameswar Panda, SouYoung Jin, Rogerio Feris, Aude Oliva, Phillip Isola, and Yoon Kim. 2024. Langnav: Language as a perceptual representation for navigation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 950–974.

Akhil Perincherry, Jacob Krantz, and Stefan Lee. 2025. Do visual imaginations improve vision-and-language navigation agents? *Preprint*, arXiv:2503.16394.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, and 1 others. 2023. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*.

Yanyuan Qiao, Wenqi Lyu, Hui Wang, Zixu Wang, Zerui Li, Yuan Zhang, Mingkui Tan, and Qi Wu. 2025. Open-nav: Exploring zero-shot vision-and-language navigation in continuous environment with open-source llms. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6710–6717. IEEE.

Yanyuan Qiao, Yuankai Qi, Zheng Yu, Jing Liu, and Qi Wu. 2023. March in chat: Interactive prompting for remote embodied referring expression. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15758–15767.

Yiwei Qin, Xuefeng Li, Haoyang Zou, Yixiu Liu, Shijie Xia, Zhen Huang, Yixin Ye, Weizhe Yuan, Hector Liu, Yuanzhi Li, and 1 others. 2024. O1 replication journey: A strategic progress report–part 1. *arXiv preprint arXiv:2410.18982*.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and 1 others. 2025. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*.

QwenLM. 2025. Qvq-72b preview.

Sonia Raychaudhuri, Saim Wani, Shivansh Patel, Unnat Jain, and Angel Chang. 2021. Language-aligned waypoint (law) supervision for vision-and-language navigation in continuous environments. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 4018–4028.

Gabriel Sarch, Yue Wu, Michael J Tarr, and Katerina Fragkiadaki. 2023. Open-ended instructable embodied agents with memory-augmented large language models. *arXiv preprint arXiv:2310.15127*.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. 2019. Habitat: A platform for embodied ai research. *Preprint*, arXiv:1904.01201.

Dhruv Shah, Błażej Osiński, Sergey Levine, and 1 others. 2023. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on robot learning*, pages 492–504. PMLR.

Yongliang Shen, Kaitao Song, Xu Tan, Wenqi Zhang, Kan Ren, Siyu Yuan, Weiming Lu, Dongsheng Li, and Yueting Zhuang. 2023. Taskbench: Benchmarking large language models for task automation. *arXiv preprint arXiv:2311.18760*.

Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, and 1 others. 2025a. Hi robot: Open-ended instruction following with hierarchical vision-language-action models. *arXiv preprint arXiv:2502.19417*.

Xiangyu Shi, Zerui Li, Wenqi Lyu, Jiatong Xia, Feras Dayoub, Yanyuan Qiao, and Qi Wu. 2025b. Smartway: Enhanced waypoint prediction and backtracking for zero-shot vision-and-language navigation. *Preprint*, arXiv:2503.10069.

Xiangyu Shi, Zerui Li, Yanyuan Qiao, and Qi Wu. 2025c. Fast-smartway: Panoramic-free end-to-end zero-shot vision-and-language navigation. *Preprint*, arXiv:2511.00933.

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

Kunal Pratap Singh, Suvaansh Bhambri, Byeonghwi Kim, Roozbeh Mottaghi, and Jonghyun Choi. 2021. Factorizing perception and policy for interactive instruction following. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1888–1897.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2998–3009.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, and 1 others. 2021. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in neural information processing systems*, 34:251–266.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, Steven Bohez, Konstantinos Bousmalis, Anthony Brohan, Thomas Buschmann, Arunkumar Byravan, Serkan Cabi, Ken Caluwaerts, Federico Casarini, Oscar Chang, and 99 others. 2025a. Gemini robotics: Bringing ai into the physical world. *Preprint*, arXiv:2503.20020.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025b. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

InternVLA-N1 Team. 2025a. InternVLA-N1: An open dual-system navigation foundation model with learned latent plans.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025c. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, and 1 others. 2024. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*.

Qwen Team. 2024. Qwq: Reflect deeply on the boundaries of the unknown.

Qwen Team. 2025b. Qwen-vl-max: Alibaba cloud's advanced large vision language model.

Omkar Thawakar, Dinura Dissanayake, Ketan More, Ritesh Thawkar, Ahmed Heakl, Noor Ahsan, Yuhao Li, Mohammed Zumri, Jean Lahoud, Rao Muhammad Anwer, and 1 others. 2025. Llamav-o1: Rethinking step-by-step visual reasoning in llms. *arXiv preprint arXiv:2501.06186*.

Sai H Vemprala, Rogerio Bonatti, Arthur Bucker, and Ashish Kapoor. 2024. Chatgpt for robotics: Design principles and model abilities. *Ieee Access*.

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

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023a. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Hanqing Wang, Wei Liang, Luc Van Gool, and Wenguan Wang. 2023b. Dreamwalker: Mental planning for continuous vision-language navigation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10873–10883.

Hanqing Wang, Wenguan Wang, Wei Liang, Caiming Xiong, and Jianbing Shen. 2021. Structured scene memory for vision-language navigation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 8455–8464.

Jiaqi Wang, Enze Shi, Huawen Hu, Chong Ma, Yiheng Liu, Xuhui Wang, Yincheng Yao, Xuan Liu, Bao Ge, and Shu Zhang. 2024a. Large language models for robotics: Opportunities, challenges, and perspectives. *Journal of Automation and Intelligence*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024b. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. 2023c. Gridmm: Grid memory map for vision-and-language navigation. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 15625–15636.

Meng Wei, Chenyang Wan, Jiaqi Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, and Xihui Liu. 2025a. Ground slow, move fast: A dual-system foundation model for generalizable vision-and-language navigation. *Preprint*, arXiv:2512.08186.

Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, Xihui Liu, and Jiangmiao Pang. 2025b. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *Preprint*, arXiv:2507.05240.

Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. 2021. Visual room rearrangement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5922–5931.

Junjie Wen, Yichen Zhu, Minjie Zhu, Jinming Li, Zhiyuan Xu, Zhengping Che, Chaomin Shen, Yaxin Peng, Dong Liu, Feifei Feng, and 1 others. 2024. Object-centric instruction augmentation for robotic manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4318–4325. IEEE.

Zhenyu Wu, Ziwei Wang, Xiuwei Xu, Jiwen Lu, and Haibin Yan. 2023. Embodied task planning with large language models. *arXiv preprint arXiv:2307.01848*.

Zhenyu Wu, Ziwei Wang, Xiuwei Xu, Jiwen Lu, and Haibin Yan. 2024. Embodied instruction following in unknown environments. *arXiv preprint arXiv:2406.11818*.

Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. 2024. Llava-cot: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*.

Haotian Xu, Xing Wu, Weinong Wang, Zhongzhi Li, Da Zheng, Boyuan Chen, Yi Hu, Shijia Kang, Jiaming Ji, Yingying Zhang, and 1 others. 2025. Redstar: Does scaling long-cot data unlock better slow-reasoning systems? *arXiv preprint arXiv:2501.11284*.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, Heng Ji, Huan Zhang, and Tong Zhang. 2025a. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *Preprint*, arXiv:2502.09560.

Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, and 1 others. 2025b. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. *arXiv preprint arXiv:2503.10615*.

Huanjin Yao, Jiaxing Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, Yuxin Song, Haocheng Feng, Li Shen, and 1 others. 2024. Mulberry: Empowering mllm with o1-like reasoning and reflection via collective monte carlo tree search. *arXiv preprint arXiv:2412.18319*.

Sriram Yenamandra, Arun Ramachandran, Karmesh Yadav, Austin Wang, Mukul Khanna, Theophile Gervet, Tsung-Yen Yang, Vidhi Jain, Alexander William Clegg, John Turner, and 1 others. 2023. Homerobot: Open-vocabulary mobile manipulation. *arXiv preprint arXiv:2306.11565*.

Naoki Yokoyama, Ram Ramrakhya, Abhishek Das, Dhruv Batra, and Sehoon Ha. 2024. Hm3d-ovon: A dataset and benchmark for open-vocabulary object goal navigation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5543–5550. IEEE.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. 2024. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Di Zhang, Jianbo Wu, Jingdi Lei, Tong Che, Jiatong Li, Tong Xie, Xiaoshui Huang, Shufei Zhang, Marco Pavone, Yuqiang Li, and 1 others. 2024a. Llama-berry: Pairwise optimization for o1-like olympiad-level mathematical reasoning. *arXiv preprint arXiv:2410.02884*.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, Yuxin Fan, Wenjun Li, Zhibo Chen, Fei Gao, Qi Wu, Zhizheng Zhang, and He Wang. 2025a. Embodied navigation foundation model. *Preprint*, arXiv:2509.12129.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. 2024b. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. *arXiv preprint arXiv:2412.06224*.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. 2024c. Navid: Video-based vlm plans the next step for vision-and-language navigation. *arXiv preprint arXiv:2402.15852*.

Lingfeng Zhang, Xiaoshuai Hao, Yingbo Tang, Haoxiang Fu, Xinyu Zheng, Pengwei Wang, Zhongyuan Wang, Wenbo Ding, and Shanghang Zhang. 2025b. *navα*Θ3: Understanding any instruction, navigating anywhere, finding anything. *arXiv preprint arXiv:2508.04598*.

Ruohong Zhang, Bowen Zhang, Yanghao Li, Haotian Zhang, Zhiqing Sun, Zhe Gan, Yinfei Yang, Ruoming Pang, and Yiming Yang. 2024d. Improve vision language model chain-of-thought reasoning. *arXiv preprint arXiv:2410.16198*.

Wenqi Zhang, Yongliang Shen, Linjuan Wu, Qiuying Peng, Jun Wang, Yueting Zhuang, and Weiming Lu. 2024e. Self-contrast: Better reflection through inconsistent solving perspectives. *arXiv preprint arXiv:2401.02009*.

Wenqi Zhang, Ke Tang, Hai Wu, Mengna Wang, Yongliang Shen, Guiyang Hou, Zeqi Tan, Peng Li, Yueting Zhuang, and Weiming Lu. 2024f. Agent-pro: Learning to evolve via policy-level reflection and optimization. *arXiv preprint arXiv:2402.17574*.

Wenqi Zhang, Hang Zhang, Xin Li, Jiashuo Sun, Yongliang Shen, Weiming Lu, Deli Zhao, Yueting Zhuang, and Lidong Bing. 2025c. 2.5 years in class: A multimodal textbook for vision-language pretraining. *arXiv preprint arXiv:2501.00958*.

Yichi Zhang and Joyce Chai. 2021. Hierarchical task learning from language instructions with unified transformers and self-monitoring. *arXiv preprint arXiv:2106.03427*.

Yuxiang Zhang, Shangxi Wu, Yuqi Yang, Jiangming Shu, Jinlin Xiao, Chao Kong, and Jitao Sang. 2024g. o1-coder: an o1 replication for coding. *arXiv preprint arXiv:2412.00154*.

Yu Zhao, Huifeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2024a. Marco-o1: Towards open reasoning models for open-ended solutions. *arXiv preprint arXiv:2411.14405*.

Zhonghan Zhao, Wenhao Chai, Xuan Wang, Boyi Li, Shengyu Hao, Shidong Cao, Tian Ye, and Gaoang Wang. 2024b. See and think: Embodied agent in virtual environment. In *European Conference on Computer Vision*, pages 187–204. Springer.

Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. 2024. Towards learning a generalist model for embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13624–13634.

Gengze Zhou, Yicong Hong, Zun Wang, Xin Eric Wang, and Qi Wu. 2024a. Navgpt-2: Unleashing navigational reasoning capability for large vision-language models. In *European Conference on Computer Vision*, pages 260–278. Springer.

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

Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. 2017. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE.

Ziyu Zhu, Xilin Wang, Yixuan Li, Zhuofan Zhang, Xiaojian Ma, Yixin Chen, Baoxiong Jia, Wei Liang, Qian Yu, Zhidong Deng, and 1 others. 2025. Move to understand a 3d scene: Bridging visual grounding and exploration for efficient and versatile embodied navigation. *arXiv preprint arXiv:2507.04047*.

Filippo Ziliotto, Tommaso Campari, Luciano Serafini, and Lamberto Ballan. 2024. Tango: Training-free embodied ai agents for open-world tasks. *arXiv preprint arXiv:2412.10402*.

Embodied-Reasoner: Synergizing Visual Search, Reasoning, and Action for Embodied Interactive Tasks

Supplementary Material

A Experiment Details

A.1 Evaluation Metrics

Metrics. Three metrics to assess trajectory quality. **Success Rate (SR)**: Whether contains all key actions in order and the final state is also correct. **Search Efficiency**: Ratio of key actions to predicted sequences ($N_{ak} \div N_{pt}$). **Task Completeness**: Proportion of completed key actions ($N_{pk} \div N_{ak}$), where N_{pk} denotes predicted key actions, N_{pt} denotes total predicted actions, and N_{ak} denotes annotated key actions.

A.2 Real-World Scenarios Experiments

To evaluate the generalization of our reasoning model, we design a real-world experiment about object searching, covering 30 tasks in three scenes. As shown in Table A7, *Embodied-Reasoner* demonstrates notable capabilities in real-world environments. In terms of success rate, it outperforms OpenAI o1 by 6.7%, OpenAI o3-mini by 12.7%, and Qwen2.5-VL-72B-Instruct by 13.4%.

A.3 Repeat Exploration Rate

The Repeat Exploration Rate (RER)indicates how often the model revisits the same location within its trajectory and is calculated as the number of revisits to previous locations divided by the total number of explorations. For example, in a task, the model navigated to the following path: $Place_a$, $Place_b$, $Place_b$, $Place_c$, $Place_c$. The model revisited $Place_b$ and $Place_c$ once each. Thus, the repeat exploration rate was 40% $(2/5)$.

As shown in Figure A1, our models (*Embodied-Reasoner/Explorer*) exhibit a lower RER (-50%) compared to baseline models across all four tasks.

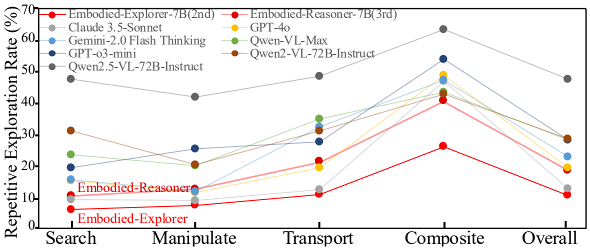


Figure A1: Repetitive Exploration Rate measures repetitive search issues, which are often observed in baselines. Our models reduce repetitive searches by recalling and reflecting on past trajectories.

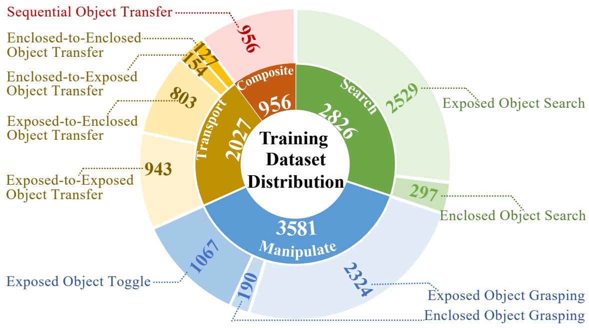


Figure B2: The distribution of the training dataset with *9,390* samples, including *4* task types and *10* sub-task types.

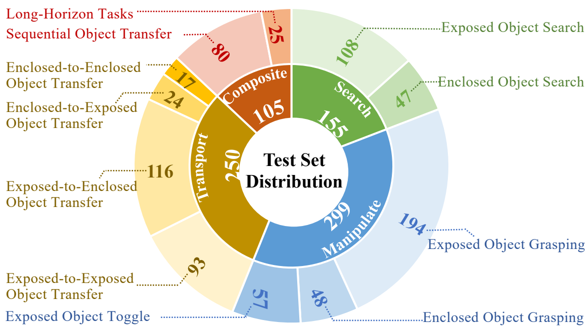


Figure B3: The distribution of the test set with *809* tasks, including *4* task types and *11* sub-task types.

B Dataset Details

B.1 Distribution of Task Instructions

We synthesize 9,390 unique task instructions along with their *Observation-Thought-Action* trajectories as the training set. The distribution of training tasks is shown in Figure B2, encompassing 4 task types(*Search*, *Manipulate*, *Transport* and *Composite*) and 10 subtask types.

For evaluation, we curate 809 test cases across 12 novel scenarios distinct from the training environments. The distribution of test tasks is shown in Figure B3, covering 4 task types and 11 corresponding subtask types.

Types	Models	Success Rate	Goal Condition	Pick & Place	Examine in Light	Pick Two & Place	Stack & Place
Imitation Learning or RL-based Expert Models	MOCA	5.36	16.18	6.00	4.60	1.20	6.40
	HiTUT	10.23	20.71	26.00	8.10	12.40	7.30
	HLSM	18.28	31.24	34.80	36.60	18.00	4.40
	FILM	20.10	32.45	16.03	29.65	11.84	1.98
	LGS-RPA	30.41	42.79	-	-	-	-
Embodied Agent	LLM-Planner+4o-mini	22.54	38.83	41.00	27.75	12.12	3.67
General VLMs	Qwen2.5-VL-72B-Instruct	35.49	51.51	50.00	42.20	30.30	14.68
	GPT-4o-mini	37.72	54.01	70.00	31.79	21.21	27.52
	Claude-3.7-Sonnet-thinking	53.41	62.36	78.00	65.90	37.29	19.44
	Gemini-2.5-Pro-thinking	57.81	66.74	84.00	61.85	60.61	25.69
Embodied Model	Gemini Robotics-ER 1.5	60.71	69.91	82.00	72.83	34.85	37.61
	Embodied-Reasoner-7B	56.03	67.72	84.00	70.52	33.33	21.10

Table A1: Detailed comparison of our model against Imitation-, RL-, and LLM-based models on ALFRED.

Model	Success Rate ↑	Search Efficiency ↑	Task Completeness ↑	Tokens (k) / Trajectory ↓
Embodied-Reasoner	**80.96%**	**55.07%**	**86.30%**	52.5
gpt-4o-mini + Memory	10.93%	5.68%	23.15%	21.5
gpt-4o + Memory	40.32%	21.56%	53.49%	12.6
Embodied-Reasoner + Memory	77.35%	42.4%	82.3%	**5.8**

Table A2: Performance of memory-augmented models. Embodied-Reasoner equipped with a memory mechanism significantly reduces consumed tokens per trajectory (from 52.5k to 5.8k, an 89% reduction) while maintaining high success rates (77.35% vs. 80.96%). Compared to GPT-4o with memory, our model achieves substantially higher performance across all metrics while using fewer tokens.

	Search	Manipulate	Transport	Composite
w/ different thinking pattern	65.16	93.31	87.20	54.29
w/o different thinking pattern	65.81 (+0.65)	88.63 (-4.68)	82.80 (-4.40)	47.52 (-6.77)

Table A3: We ablate the five thinking patterns by using only a single <thinking> tag, which shows significant performance drops on more challenging Transport and Composite tasks.

Training Stages			Success Rate ↑	Search Efficiency ↑	Task Completeness ↑	Success Rate for SubTasks ↑			
1st Stage	2nd Stage	3rd Stage				Search	Manipulate	Transport	Composite
✓	✓	✓	**81.0%**	**55.1%**	**86.3%**	**65.2%**	**93.3%**	**87.2%**	**54.3%**
✗	✓	✓	19.1%	11.6%	43.6%	17.8%	22.0%	24.2%	0.9%
✓	✗	✓	52.1%	27.8%	65.1%	35.2%	68.3%	61.3%	9.4%
✓	✓	✗	65.4%	46.3%	77.7%	60.0%	75.9%	72.2%	26.7%

Table A4: Ablation experiment on three training stages. We observe that without the first stage of imitation learning using synthetic trajectories, the model struggles to learn basic interaction behaviors and cannot generate sufficient samples for subsequent rejection sampling and reflection tuning. Without the second and third stages, the model performs significantly worse on composite tasks.

B.2 Distribution of Interaction Types

In the training set, trajectories consist of eight types of interaction actions: *navigate to, pickup, open, close, put in, observe, move forward, and toggle*. As shown in the Figure B4, the occurrence frequency of each interaction action across all trajectories is illustrated. Among them, the exploration action *navigate to* appears the most frequently, occurring over 29k times.

In the test set, we manually design instructions and annotate the corresponding key actions and final states. The test tasks involve six types of interactions: *navigate to, pickup, open, close, put in, and toggle*. As seen in the Figure B5, *navigate to* also appears significantly more frequently than other key actions.

B.3 Distribution of Task Length

In the training set, each trajectory consists of an average of 7.2 interactions with the environment (e.g., *navigate*, *pickup*). For the four task types: *Search*, *Manipulate*, *Transport*, and *Composite*. Due to varying task complexity, the corresponding trajectory lengths also differ. As shown in the Figure B6, *Search* tasks tend to have shorter trajectories, typically ranging from 1 to 9, while the more complex

Model	Training Data	R2R Val Unseen	
		SR	SPL
NavGPT(Zhou et al., 2024b)	zero-shot	34.0	29.0
DiscussNav(Long et al., 2023)	zero-shot	43.0	40.0
NavCoT(Lin et al., 2025)	In-domain data from Matterport3D	40.2	36.6
NaVILA(Cheng et al., 2024)	In-domain data from Matterport3D	54.0	49.0
StreamVLN(Wei et al., 2025b)	In-domain data from Matterport3D	56.9	51.9
InternVLA-N1(Team, 2025a)	In-domain data from Matterport3D	58.2	54.0
NavFoM(Zhang et al., 2025a)	In-domain data from Matterport3D	61.7	55.3
DualVLN(Wei et al., 2025a)	In-domain data from Matterport3D	64.3	58.5
Embodied-Reasoner	Out-of-domain	55.0	42.8

Table A5: Comparison with state-of-the-art VLN models on R2R benchmark. Our model demonstrates competitive performance on the challenging vision-language navigation task.

	Setting	Avg	Base	Common	Complex	Visual	Spatial	Long
GPT-4o-mini	10-shot	32.7	74	22	32	22	32	14
Llama-3.2-11B-Vision(Meta AI, 2024)	10-shot	25.0	70	16	28	10	20	6
InternVL2_5-8B(Chen et al., 2025)	10-shot	11.3	36	4	0	10	16	2
Qwen2-VL-72B-Instruct	10-shot	35.7	70	30	36	32	28	18
Qwen2.5-VL-72B-Instruct	10-shot	37.7	74	28	42	40	24	18
gemma-3-12b-it(Team et al., 2025b)	10-shot	23.0	58	10	24	18	24	4
Embodied-Reasoner-7B	zero-shot	37.7	66	36	26	46	24	28

Table A6: Comparison with state-of-the-art VLMs on EmbodiedBench (EB-Habitat). Our model demonstrates strong zero-shot performance, outperforming larger models with 10-shot prompting.

Model	**Success Rate (%)**
Qwen2.5-VL-72B-Instruct	43.3
OpenAI o1	50.0
OpenAI o3-mini	43.4
Embodied-Reasoner	56.7

Table A7: Real-world experiments results: Embodied-Reasoner outperforms OpenAI o1 by 6.7% in success rate, showing notable real-world capability.

Composite tasks generate the longest trajectories, usually exceeding 8 and reaching beyond 23.

Similarly, in the test set, as shown in the Figure B7, the more complex *Composite* tasks also exhibit the longest key action sequences, usually exceeding 8 and reaching beyond 19.

B.4 Distribution of Object Categories

Our training dataset comprises 107 diverse indoor scenes(kitchens, living rooms, etc.), over 2,100 interactive objects (e.g., eggs, laptops) and 2,600 containers (e.g., refrigerators, drawers). Across the 9,390 unique task instructions, trajectories involve interactions with these objects and containers, with the top 32 most frequently explored and interacted object categories shown in Figure B8. The 12 distinct test set scenes feature key actions tied to different objects, with the top 32 frequent containers and interactive objects illustrated in Figure B9.

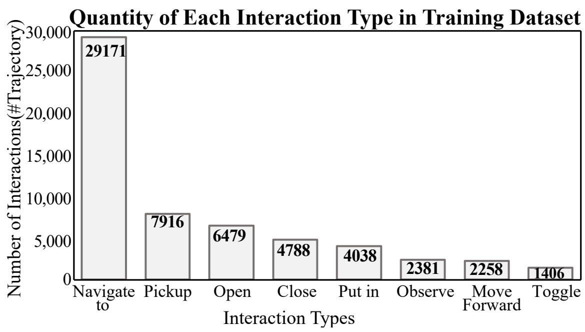


Figure B4: The distribution of the training set interactions, including *8* interaction types in trajectories: *navigate to, pickup, open, close, put in, observe, move forward, and toggle*.

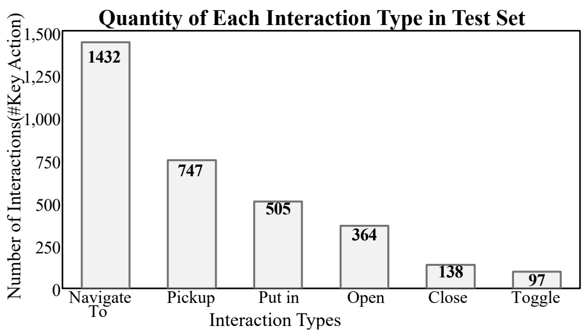


Figure B5: The distribution of the test set interactions, including *6* interaction types in key actions: *navigate to, pickup, open, close, put in, and toggle*.

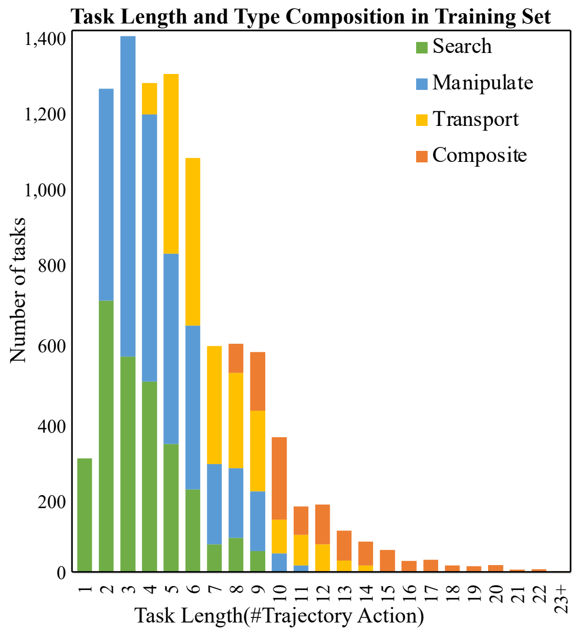


Figure B6: The quantity distribution of trajectory lengths in the training set and the corresponding task type composition, where *Search Task* is mainly within *1-9*, *Manipulate Task* within *2-11*, *Transport Task* within *3-14*, and *Composite Task* above *8*, extending beyond 23.

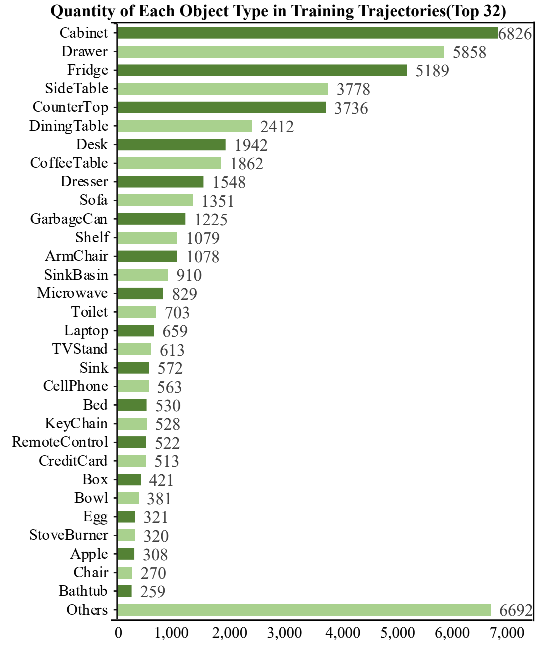


Figure B8: The quantity distribution of the top 32 object types in the training dataset's trajectories, with *Others* representing the remaining 62 categories, such as *Bread, Book, DeskLamp*, etc.

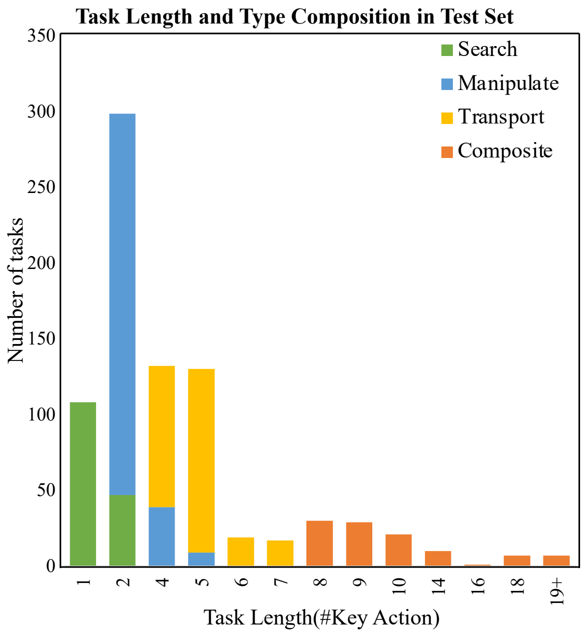


Figure B7: The quantity distribution of key action lengths in the test set and the corresponding task type composition, where *Search Task* is mainly within 1-2, *Manipulate Task* within 2, 4-5, *Transport Task* within 4-7, and *Composite Task* above 8, extending beyond 19.

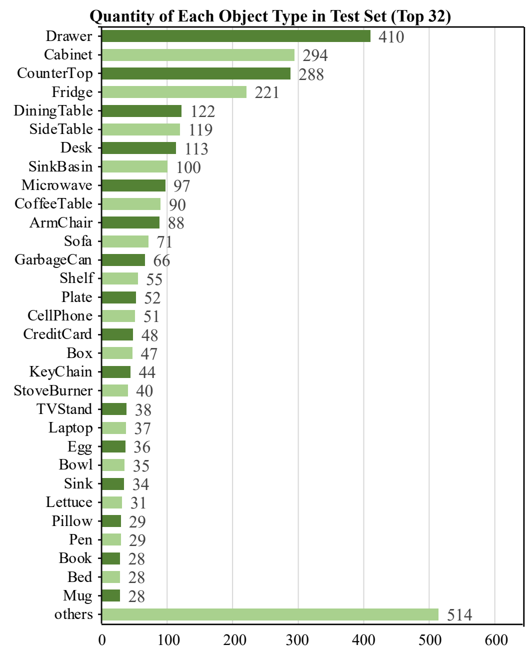


Figure B9: The quantity distribution of the top 32 object types in the test set's key actions, with *Others* representing the remaining 44 categories, such as *Watch, Pencil, Cup*, etc.

B.5 Detailed Task Templates and Constraints

Our four daily tasks: *Search*, *Manipulate*, *Transport*, and *Composite* can be further divided into corresponding sub-tasks based on the types of objects involved. We design multiple task templates for each task type. It ensures synthesized instruction's validity. Templates and constraints are shown in Table B8 and Table B9.

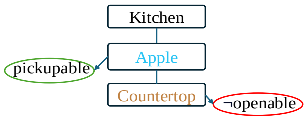
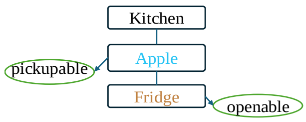
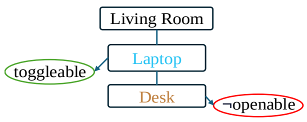
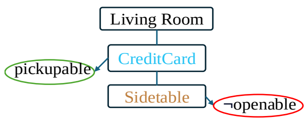
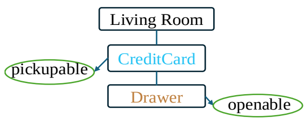
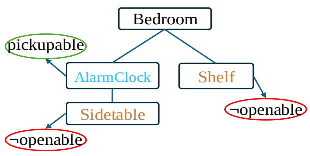
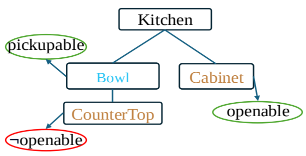
Task Types	Sub-Task Types	Templates	Constraint Check	Case	Affiliation and Attribute
Search	Exposed Object Search	find A	Pickupable(A) ∧ ¬ Openable(Parent(A))	**Task:** Could you please find the Apple in the room? **Key Action Sequences:** navigate to CounterTop end	
	Enclosed Object Search	find A	Pickupable(A) ∧ Openable(Parent(A))	**Task:** Could you please find the Apple in the room? **Key Action Sequences:** navigate to Fridge open Fridge end	
Manipulate	Exposed Object Toggle	toggle A	Toggleable(A) ∧ ¬ Openable(Parent(A))	**Task:** Would you mind powering on the Laptop for me? **Key Action Sequences:** navigate to Desk toggle Laptop, end	
	Exposed Object Grasping	pickup A	Pickupable(A) ∧ ¬ Openable(Parent(A))	**Task:** I want to pick up a CreditCard from the room, can you help me? **Key Action Sequences:** navigate to SideTable pickup CreditCard end	
	Enclosed Object Grasping	pickup A	Pickupable(A) ∧ Openable(Parent(A))	**Task:** Would it be possible for you to pick up a CreditCard from the room? **Key Action Sequences:** navigate to Drawer open Drawer pickup CreditCard, end	
Transport	Exposed-to-Exposed Object Transfer	pickup A put in B	Pickupable(A) ∧ ¬ Openable(Parent(A)) ∧ ¬ Openable(B)	**Task:** Could you please put the AlarmClock on the Shelf? **Key Action Sequences:** navigate to Sidetable pickup AlarmClock navigate to Shelf put in Shelf, end	
	Exposed-to-Enclosed Object Transfer	pickup A put in B	Pickupable(A) ∧ ¬ Openable(Parent(A)) ∧ Openable(B)	**Task:** Would you mind placing the Bowl in the Cabinet, please? **Key Action Sequences:** navigate to CounterTop pickup Bowl navigate to Cabinet open Cabinet put in Cabinet, end	

Table B8: Details of task templates for four task types

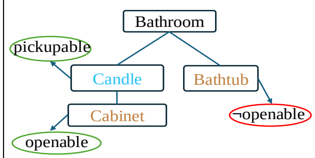
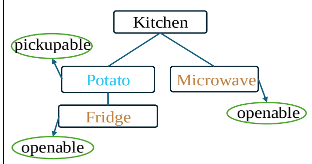
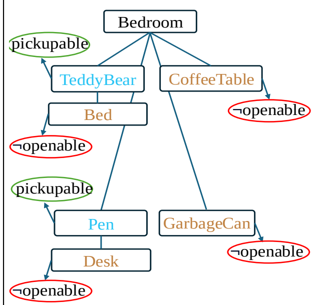
Task Types	Sub-Task Types	Templates	Constraint Check	Case	Affiliation and Attribute
Transport	Enclosed-to-Exposed Object Transfer	pickup A put in B	Pickupable(A) ∧ Openable(Parent(A)) ∧ ¬ Openable(B)	**Task:** Is it okay to put the Candle on the Bathtub? **Key Action Sequences:** navigate to Cabinet open Cabinet pickup Candle close Cabinet navigate to Bathtub put in Bathtub end	
	Enclosed-to-Enclosed Object Transfer	pickup A put in B	Pickupable(A) ∧ Openable(Parent(A)) ∧ Openable(B)	**Task:** May I ask you to put the Potato in the Microwave? **Key Action Sequences:** navigate to Fridge open Fridge pickup Potato close Fridge navigate to Microwave open Microwave put in Microwave end	
Composite	Sequential Object Transfer	first, pickup A1 put in B1 then, pickup A2 put in B2	Pickupable(A1) ∧ ¬ Openable(Parent(A1)) ∧ Openable(B1) ∧ Pickupable(A2) ∧ ¬ Openable(Parent(A2)) ∧ Openable(B2) ∧ Different(A1, A2)	**Task:** Could you please first place the TeddyBear on the CoffeeTable, and then place the Pen on the GarbageCan? **Key Action Sequences:** navigate to Bed pickup TeddyBear navigate to CoffeeTable put in CoffeeTable navigate to Desk pickup Pen navigate to GarbageCan put in GarbageCan end	
	Long-Term Complex Task	——	——	**Task:** Could you put the bread in the refrigerator and then take the apple out of the refrigerator, wash it, and place it on a plate? **Key Action Sequences:** navigate to CounterTop, pickup Bread, navigate to Fridge, open Fridge put in Fridge, pickup Apple navigate to SinkBasin, put in SinkBasin toggle Facuet, pickup Apple navigate to Cabinet, open Cabinet put Plate, close Cabinet end	

Table B9: Details of task templates for four task types

Scene Type	Task	Key Action Sequences
Kitchen	Could you kindly place the coffee mug in the refrigerator when you have a moment?	navigate to countertop→ pickup coffee mug→ navigate to fridge→ open fridge→ put in fridge→ end
	Would you mind putting the umbrella on the kitchen island counter whenever convenient?	navigate to coffee table→ pickup umbrella→ navigate to countertop→ put in countertop→ end
	If it's not too much trouble, could you please put the glass cup in the microwave?	navigate to diningtable→ pickup glass cup→ navigate to microwave→ open microwave→ put in microwave→ end
	I'd appreciate if you could leave the milk on the coffee table when possible.	navigate to fridge→ open fridge→ pickup milk→ navigate to coffee table→ put in coffee table→ end
	Whenever you get a chance, would you be so kind as to place the coffee cup in the microwave?	navigate to cabinet→ open cabinet→ pickup coffee cup→ navigate to microwave→ open microwave → put in microwave→ end
	If you don't mind, could we first tuck the umbrella into the backpack, then store the glass cup in the fridge, and finally place the milk in the microwave at your earliest convenience?	navigate to coffee table→ pickup umbrella→ navigate to armchair→ put in backpack→ navigate to diningtable→ pickup glass cup→ navigate to fridge → open fridge→ put in fridge→ pickup milk→ navigate to microwave→ open microwave→ put in microwave → end
Bathroom	Could you please place the cola in the cabinet?	navigate to shelf→ pickup cola → navigate to cabinet → open cabinet → put in cabinet → end
	Would you mind setting the remote control on the shelf?	navigate to cabinet→ open cabinet→ pickup remote control→ navigate to shelf→ put in shelf→ end
	Could you kindly put the coffee cup on the desk?	navigate to shelf→ pickup coffee cup→ navigate to desk → put in desk→ end
	Would you be so kind as to pick up the coffee cup?	navigate to shelf→ pickup coffee cup→ end
	If you don't mind, could you place the glass cup on the chair?	navigate to desk→ pickup glass cup→ navigate to chair → put in chair→ end
	Could you please set the tissues on the chair?	navigate to desk→ pickup tissues→ navigate to chair → put in chair→ end
	If it's not too much trouble, could you place the book on the desk?	navigate to bookshelf→ pickup book→ end
	Would you mind picking up the book from the room?	navigate to bookshelf→ pickup book→ navigate to desk → put in desk→ end
	Would you kindly put the glass cup inside the box?	navigate to desk→ pickup glass cup→ navigate to box → put in box→ end
	Could you please place the pen in the cabinet?	navigate to bookshelf→ pickup pen→ navigate to cabinet → put in cabinet→ end
	If possible, could you first put the glass cup in the box, then put the cola, and finally set the box on the shelf?	navigate to desk→ pickup glass cup→ navigate to box → put in box→ navigate to shelf→ pickup cola → navigate to box→ put in box→ pickup box → navigate to shelf→ put in shelf→ end
	Would you mind placing the cola in the trash can first, then putting the credit card on the chair, and finally putting the trash can on the shelf?	navigate to shelf→ pickup cola→ navigate to trash can → put in trash can→ navigate to desk→ pickup credit card → navigate to chair→ put in chair→ navigate to trash can → pickup trash can→ navigate to shelf→ put in shelf→ end
Bedroom	Could you please get the towel from the room?	navigate to towel holder→ pickup towel→ end
	Would you mind getting the roll of toilet paper?	navigate to roll of toilet paper holder→ pickup roll of toilet paper→ end
	If it's not too much trouble, could you place the body wash bottle in the trash can?	navigate to shelf→ pickup body wash bottle→ navigate to trash can→ put in trash can→ end
	Would you kindly put the soap in the sinkbasin?	navigate to shelf→ pickup soap→ navigate to sinkbasin → put in sinkbasin→ end
	Could you please find the glass cup and pick it up?	navigate to cabinet→ open cabinet→ pickup glass cup → end
	If it's convenient, could you pick up the hair dryer bottle?	navigate to shelf→ pickup hair dryer bottle→ end

Table B10: Real world task instructions: 30 Object searching tasks across multiple real scenes(6 kitchen, 12 bathroom, 12 bedroom)

Scene Type	Task	Key Action Sequences
Bedroom	Would you mind putting the soap on the countertop?	navigate to shelf→ pickup soap→ navigate to countertop → put in countertop→ end
	Could you kindly place the towel in the sinkbasin?	navigate to towel holder→ pickup towel→ navigate to sinkbasin→ put in sinkbasin→ end
	If you don't mind, could you first put the body wash bottle in the sinkbasin and then place the soap in the trash can?	navigate to shelf→ pickup body wash bottle→ navigate to sinkbasin→ put in sinkbasin→ navigate to shelf→ pickup soap→ navigate to trash can→ put in trash can→ end
	Would you be so kind as to put the body wash bottle in the cabinet?	navigate to shelf→ pickup body wash bottle→ navigate to cabinet→ open cabinet→ put in cabinet→ end
	First, place the roll of toilet paper on the shelf. Then, put the body wash bottle on the countertop. Finally, put the soap in the sinkbasin.	navigate to roll of toilet paper holder→ pickup roll of toilet paper→ navigate to shelf→ put in shelf→ pickup body wash bottle→ navigate to countertop→ put in countertop→ navigate to shelf→ pickup soap→ navigate to sinkbasin→ put in sinkbasin→ end
	Close the toilet lid. Next, place the soap on the countertop. And finally, put the body wash bottle in the sinkbasin.	navigate to toilet→ close toilet→ navigate to shelf→ pickup soap→ navigate to countertop→ put in countertop → navigate to shelf→ pickup body wash bottle→ navigate to sinkbasin→ put in sinkbasin→ end

Table B11: Real-world task instructions: 30 Object searching tasks across multiple real scenes(6 kitchen, 12 bathroom, 12 bedroom)

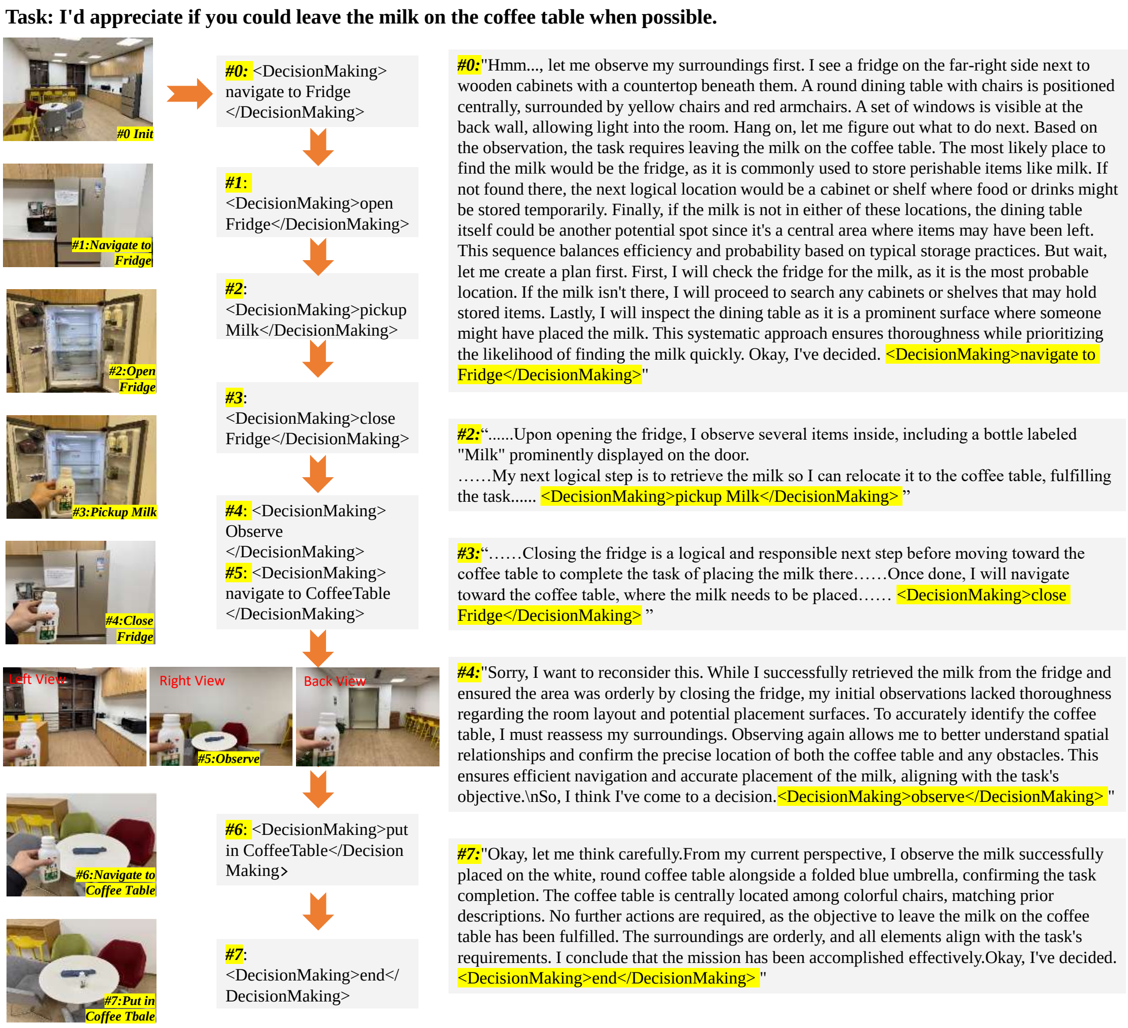


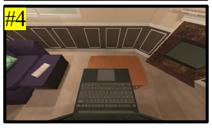
Figure B10: Trajectory case for Embodied Reasoner in real world.

TASKNAME: Could you please first place the Laptop on the Sofa, and then place the CellPhone on the Drawer?

Hmm..., let me observe my surroundings first. I see a Sofa positioned near a CoffeeTable at the center of the room….**My plan is to search for the Laptop on the SideTable first, as it is likely to hold such an object and aligns with the current subtask requirements**. Once located, I will proceed to place the Laptop on the Sofa to complete the first subtask. If the Laptop is found here, further actions will be planned accordingly based on subsequent observations. So, I think I've come to a decision**.<DecisionMaking>navigate to SideTable</DecisionMaking>**

Hold on, I need to rethink this.From the images, the room contains several surfaces: a CoffeeTable with objects on top, a Sofa with items placed on it, and other furniture. The Laptop is still unfound despite prior efforts. ,…,Aligning with my goal to locate the Laptop and place it on the Sofa, focusing on the CoffeeTable as a potential location is both practical and consistent with the task requirements. Thus, I will navigate to the CoffeeTable to continue my search.Hmm..., I think I know what to do.**<DecisionMaking>navigate to CoffeeTable</DecisionMaking>**

Wait up, lemme get my thoughts together. In the updated view, I notice a Laptop placed prominently on a wooden table near the Sofa… The Sofa, where the Laptop needs to be placed, is conveniently adjacent to this location... **My next step is to pick up the Laptop from the table.** After securing the object, I will proceed to place it on the Sofa to fulfill the current subtask. This sequential approach ensures efficient progress toward achieving the objective. Okay, I've decided.**<DecisionMaking>pickup Laptop</DecisionMaking>**



Hang on, let me figure out what to do next. From my current view, I confirm that I am holding the Laptop securely…As the Sofa is the designated target for this subtask, moving closer to it will allow me to complete the action efficiently without any complications. **But wait, let me create a plan first**. I will navigate to the Sofa, ensuring that I position myself optimally to place the Laptop onto it. Completing this action will advance me toward fulfilling the current subtask. Hmm..., I've settled on a choice**.<DecisionMaking>navigate to Sofa</DecisionMaking>**

Okay, I think I'll start by making a search plan. In the image, I see the Sofa clearly in front of me, and I am holding the Laptop. The Sofa is unobstructed, providing an ideal surface to place the Laptop. To complete the current subtask, I will carefully position the Laptop on the Sofa, ensuring stability and alignment with the task requirements. This action progresses the task efficiently and prepares me for the subsequent objective. Okay, I've decided**.<DecisionMaking>put in Sofa</DecisionMaking>**

In the image, I observe a well-lit room with various furniture pieces. The Sofa, where I just placed the Laptop, is prominent in the foreground.…**For my current subtask placing the CellPhone on the Drawer I need to locate the CellPhone first**. Considering the logical placement of small objects, the SideTable appears to be a suitable candidate to explore next. Therefore, I will navigate to the SideTable to search for the CellPhone. Hmm..., I've settled on a choice**.<DecisionMaking>navigate to SideTable</DecisionMaking>**

Sorry, let me think about this a bit longer.In the image, I observe a SideTable with a plant and other items but no sign of the CellPhone. **Reflecting on my existing plan, searching for the CellPhone logically continues with exploring plausible surfaces.** The DiningTable, a common location for small objects like phones, stands out as a suitable next step. …Okay, I think I've made up my mind**.<DecisionMaking>navigate to DiningTable</DecisionMaking>**

Hmm..., I think I'll make a plan first. In the image, I observe a DiningTable with several objects on it, including a CellPhone positioned near the edge. This matches the second target object required for the current subtask. **As the CellPhone is clearly visible and accessible, my immediate plan is to pick it up.** Securing the object is essential for progressing toward the goal of placing it on the Drawer. Once picked up, I will determine the optimal path to reach the Drawer and finalize this subtask. Hmm..., I think I know what to do**.<DecisionMaking>pickup CellPhone</DecisionMaking>**

But wait, let me create a plan first. In the image, I observe a DiningTable with chairs surrounding it and several objects on its surface, including a CellPhone, which I have already picked up,…, I will navigate to the Dresser, as it is a typical location for drawers and aligns with the task requirements. Hmm..., I've settled on a choice**.<DecisionMaking>navigate to Drawer</DecisionMaking>**

Hmm..., I think I'll make a plan first. In the image, I observe a Dresser with multiple Drawers beneath its surface. The CellPhone I am holding needs to be placed inside the Drawer. To achieve this, I must first open the Drawer to create access for placement. My next step is to interact with the Drawer to open it, enabling me to proceed with placing the CellPhone inside and completing the subtask efficiently. Hmm..., I think I know what to do**.<DecisionMaking>open Drawer</DecisionMaking>**

Hmm..., let me think about what to do next. In the image, I observe the open Drawer of the Dresser in front of me. ,…,wait, let me create a plan first. My next step is to carefully place the CellPhone inside the open Drawer. This action will conclude the current subtask and ensure proper alignment with the task's objectives. Following this, I will verify that the action has been executed successfully before proceeding to any subsequent tasks. Okay, I think I've made up my mind**.<DecisionMaking>put in Drawer</DecisionMaking>**

Hmm..., let me think about what to do next.I observe the open Drawer with the CellPhone placed inside, completing the specified subtask**. Earlier, I successfully placed the Laptop on the Sofa as required. Both subtasks outlined in the overall objective have been fulfilled.** With no additional tasks evident, I conclude that the entire assignment has been accomplished effectively.Okay, I think I've made up my mind**.<DecisionMaking>end</DecisionMaking>**

Figure B11: Trajectory case for Embodied Reasoner.

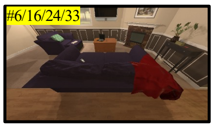
TASKNAME: Could you please first place the Laptop on the Sofa, and then place the CellPhone on the Drawer?

2.<DecisionMaking>navigate to Laptop</DecisionMaking>

<|feedback|>Action: navigate to Laptop is illegal, Laptop is the most relevant item in this room and navigate to Laptop. Object: Laptop is not currently navigable, you can try "navigate to <object>" to reach nearby, larger objects for closer observation.

3.<DecisionMaking>navigate to DiningTable</DecisionMaking>
4.<DecisionMaking>pickup Laptop</DecisionMaking>

<|feedback|>Action: pickup Laptop is illegal, Object: Laptop is currently unavailable for interaction. Possible situations include: Laptop does not exist in your current view; you are too far away from Laptop; the Laptop cannot perform operation pickup.
You can try "move forward" to approach the target object or "navigate to <object>" to reach nearby, larger objects for closer inspection.

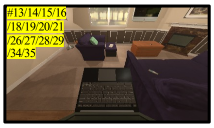


5.<DecisionMaking>move forward</DecisionMaking>
6.<DecisionMaking>navigate to Sofa</DecisionMaking>
7.<DecisionMaking>navigate to CoffeeTable</DecisionMaking>
8.<DecisionMaking>pickup Laptop</DecisionMaking>
9.<DecisionMaking>put in Sofa</DecisionMaking>

|feedback|>Action: put in Sofa is illegal, Object: Sofa is currently unavailable for interaction. Possible situations include: Sofa does not exist in your current view; you are too far away from Sofa; the Sofa cannot perform operation Put in.
You can try "move forward" to approach the target object or "navigate to <object>" to reach nearby, larger objects for closer inspection.

10.<DecisionMaking>navigate to Sofa</DecisionMaking>
11.<DecisionMaking>move forward</DecisionMaking>
12.<DecisionMaking>observe</DecisionMaking>
13.<DecisionMaking>move forward</DecisionMaking>
14.<DecisionMaking>move forward</DecisionMaking>

15.<DecisionMaking>move forward</DecisionMaking>
16.<DecisionMaking>move forward</DecisionMaking>
17.<DecisionMaking>navigate to Sofa</DecisionMaking>
18.<DecisionMaking>move forward</DecisionMaking>



19.<DecisionMaking>move forward</DecisionMaking>
20.<DecisionMaking>move forward</DecisionMaking>
21.<DecisionMaking>move forward</DecisionMaking>
22.<DecisionMaking>navigate to Chair</DecisionMaking>
23.<DecisionMaking>put in DiningTable</DecisionMaking>
24.<DecisionMaking>pickup Laptop</DecisionMaking>
25.<DecisionMaking>navigate to Sofa</DecisionMaking>
26.<DecisionMaking>move forward</DecisionMaking>
27.<DecisionMaking>move forward</DecisionMaking>
28.<DecisionMaking>move forward</DecisionMaking>
29.<DecisionMaking>move forward</DecisionMaking>
30.<DecisionMaking>navigate to CoffeeTable</DecisionMaking>
31.<DecisionMaking>move forward</DecisionMaking>

32.<DecisionMaking>observe</DecisionMaking>
33.<DecisionMaking>navigate to Chair</DecisionMaking>
34.<DecisionMaking>navigate to Sofa</DecisionMaking>
35. <DecisionMaking>move forward</DecisionMaking>
36.<DecisionMaking>move forward</DecisionMaking>

Figure B12: Trajectory case for GPT-o1.

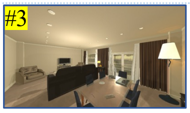
TASKNAME: Could you put the Pillow in the ArmChair?

Summarized Trajectories: []
Image buffer: []

Okay, just give me a moment to think about it. To complete the task, I need to follow these steps: Locate the "Pillow" in the environment. Navigate to the "Pillow". Pick up the "Pillow". Navigate to the "ArmChair". Put the "Pillow" in the "ArmChair". Since I don't immediately see a "Pillow" in my frontal perspective, I'll start by observing my surroundings to locate it...<DecisionMaking>observe</DecisionMaking>

Summarized Trajectories: You entered the room and see image1.
Image buffer: [#1]

Hang on, let me figure out what to do next. In the current image (image2), I still don't see the pillow, so I need to gather more information. Based on the views I now have, I suspect there might be more information available if I move forward... <DecisionMaking>move forward</DecisionMaking>

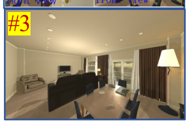


Summarized Trajectories: … image1 … After successfully executing your previous action observe, you received image2… you see an armchair on the left, but you cannot locate the pillow…
Image buffer: [#1, #2]

Hold on, I want to think this over. From the current observations and the history, I still haven‘t identified the pillow’s location. Thus, the next logical step is to explore further by observing alternate sides to locate the pillow. My thinking: I should use the "observe" action to check my surroundings for the pillow... <DecisionMaking>observe</DecisionMaking>

Summarized Trajectories: … image1 …observe, you received image2… . After successfully executing your previous action "move forward", you received image3… its location is still unknown …
Image buffer: [#1, #2, #3]

Okay, let me think carefully. Based on the new image4, there is still no clear view of a pillow in the visible surroundings. To locate the pillow effectively, I need to gather more information. Reviewing previous images could help confirm potential locations or areas I might have overlooked… <DecisionMaking>view image3</DecisionMaking>



Summarized Trajectories: … image1 …observe, you received image2… move forward, you received image3… After successfully executing your previous action "observe," you received image4…there is still no clear view of a pillow in the visible surroundings… your next action is to "view image3"
Image buffer: [#1, #2 , #3 , #4]

Hold on, I need to rethink this. Based on the given history and available images, I have not yet identified the location of the pillow. In the current image and recent images, the pillow is not visible either on the armchair or elsewhere in the room. To continue my search and inspect the surroundings more effectively, I will navigate to a nearby object for a closer look to locate the pillow. Thinking systematically, navigating closer to objects in the scene such as the sofa might reveal the pillow's presence… <DecisionMaking>navigate to sofa</DecisionMaking>

Summarized Trajectories: … image1 …observe, you received image2… move forward, you received image3… …observe, you received image4…
Image buffer: [#1, #2 , #3, #4]

Okay, let me think carefully. The pillow is now clearly visible on the sofa in the current image (image5). The next step is to pick it up so it can be placed in the armchair… <DecisionMaking>pickup Pillow</DecisionMaking>

Summarized Trajectories: … image1 …observe, you received image2… … you see an armchair on the left… move forward, you received image3… …observe, you received image4… …view image3… . After successfully executing navigate to Sofa, you received image5, which shows the pillow clearly visible on the sofa.
Image buffer: [#1, #2 , #3, #4, #5]

Okay, just give me a moment to think about it. I have already picked up the pillow, and the armchair was previously seen on the left side in image2. I will now navigate to the armchair to correctly place the pillow…<DecisionMaking>navigate to ArmChair</DecisionMaking>

Summarized Trajectories: … image1 …image2…image3… image4…view image3…image5..After picking up the Pillow, you received image6...
Image buffer: [#1, #2 , #3, #4, #5, #6]

Wait, let me think...The armchair is within reach, and the pillow is in my possession based on the task and image. I should now put the pillow into the armchair to complete the task…<DecisionMaking>put in ArmChair</DecisionMaking>

Summarized Trajectories: … image1 … image2…image3… image4…view image3…image5…image6 …After successfully executing navigate to ArmChair, you received image7. You are now in front of the armchair with the pillow in your possession…
Image buffer: [#1, #2 , #3, #4, #5, #6, #7]

…Based on the current image and context, I see that the task has already been completed successfully. The pillow is already placed in the armchair, as shown in the image. There is no further action required… <DecisionMaking>end</DecisionMaking>

Figure B13: Trajectory case for Embodied-Reasoner with memory mechanism. The model receives compressed textual summaries instead of full interleaved image-text context, significantly reducing context length. At the 4th interaction, the model actively retrieves a previous observation image from the image buffer by issuing the action `view image3`, demonstrating its ability to selectively access historical visual information when needed.

C Related Works

Previously, many researchers utilized imitation learning to mimic expert demonstrations or reinforcement learning for embodied learning, e.g., vision-language navigation (Huang et al., 2023a; Ziliotto et al., 2024; Yokoyama et al., 2024; Khanna et al., 2024; Ehsani et al., 2024; Zhu et al., 2025; Zhou et al., 2024b). In the era of LLMs, they has increasingly combined LLMs as planners to decompose natural language instructions into executable primitive actions (Wu et al., 2023; Liang et al., 2023; Song et al., 2023; Sarch et al., 2023; Huang et al., 2022b; Majumdar et al., 2024; Li et al., 2024) or end-to-end vision-language-action models for direct physical interaction (Team et al., 2024; Black et al., 2024; Shi et al., 2025a). However, existing embodied approaches often focus on single skills (manipulation or navigation) and neglect explicit reasoning scenarios for complex, long-horizon tasks involving environmental exploration, object interaction, and coherent planning. Please refer to Section C for a detailed related works.

C.1 LLM and Its Reasoning

Recent reasoning models, e.g., OpenAI-o1, DeepSeek-R1, (Guo et al., 2025; OpenAI, 2024c; DeepMind, 2025; Zhang et al., 2024g; Zhao et al., 2024a; Team, 2024; Wang et al., 2022; Zelikman et al., 2022; Shen et al., 2023) have demonstrated powerful reasoning capabilities, significantly enhancing their ability on college-level questions such as mathematics. Unlike previous efforts to scale up training data and model sizes, these systems involve generating long chain-of-thought (CoT) during inference time, improving the performance of final answers (Guan et al., 2025; Min et al., 2024; Zhang et al., 2024a,e,f; Qin et al., 2024). Besides, many efforts (Thawakar et al., 2025; Yang et al., 2025b; Huang et al., 2025a; Xu et al., 2024, 2025; Zhang et al., 2024d), such as QVQ (QwenLM, 2025) and Kimi-1.5 (Team et al., 2025c), have extended the deep-thinking paradigm to multimodal scenarios by post-training on long-CoT or reinforcement learning with verifiable rewards. However, most visual reasoning models operate in a single-round dialogue setting: processing input images and user's query and then generating textual thoughts for a final answer. This limits their applicability in embodied interactive tasks (Kolve et al., 2017; Deitke et al., 2022; Yenamandra et al., 2023; Weihs et al., 2021; Ehsani et al., 2021; Li et al., 2023), which require handling multi-image or image-text interleaved contexts (Zhang et al., 2025c) while generating coherent and logical thoughts. Besides, embodied scenes differ from mathematical tasks, as they demand long-horizon planning and deliberate reasoning over previous trajectories. In this paper, we propose an effective solution to extend general VLMs into embodied reasoning models.

C.2 Embodied Reasoning and Acting

As the language understanding capabilities of LLMs improve, recent researches integrate them into embodied tasks (Ahn et al., 2022; Huang et al., 2022c; Vemprala et al., 2024; Lin et al., 2023; Wake et al., 2024; Wang et al., 2023a; Driess et al., 2023; Huang et al., 2022b; Liang et al., 2023; Song et al., 2023; Sarch et al., 2023; Wang et al., 2024a; Majumdar et al., 2024; Azzolini et al., 2025; Dasgupta et al., 2023). Their core idea mostly involves leveraging LLMs as the "brain" or planner to decompose natural language instructions into a series of executable primitive actions or code invocations (Wu et al., 2024; Zhao et al., 2024b; Kannan et al., 2024; Mu et al., 2023; Huang et al., 2023b; Cai et al., 2024; Wu et al., 2023). This design leverages the rich commonsense and task planning abilities of LLMs without involving specific action execution. Furthermore, beyond serving as planners, researchers explore transforming LLMs/VLMs into end-to-end vision-language-action models (VLA) (Qu et al., 2025; Wen et al., 2024; Zawalski et al., 2024; Team et al., 2024; Shi et al., 2025a), enabling them to act in physical world, e.g., RT-2 (Brohan et al., 2023), OpenVLA (Kim et al., 2024), PI-0 (Black et al., 2024). Unlike VLA models, *Embodied-Reasoner* focuses on leveraging language CoT for task decomposition, environmental exploration, and visual reasoning, without involving specific action execution. This decoupled design of high-level reasoning and low-level action execution enables robots to better accomplish complex, long-horizon embodied interaction tasks.

C.3 Vision-Language Navigation

Another embodied task similar to ours is Vision-Language Navigation (VLN), which requires agents to navigate through unknown environments by following natural language instructions or image references (Krantz et al., 2020; Chen et al., 2021a,b; Wang et al., 2021; Raychaudhuri et al., 2021; Hong et al., 2022; Huang et al., 2022a; Chen et al., 2022b,a; Georgakis et al., 2022; Wang et al., 2023b; Huang et al., 2023a; Shah et al., 2023; Qiao et al., 2023; Chen et al., 2023; Wang et al., 2023c; Zheng et al., 2024; Pan et al., 2024; An et al., 2024; Zhang et al., 2024c,b; Ziliotto et al., 2024; Zhou et al., 2024b,a; Chen et al., 2024; Zhu et al., 2025; Wei et al., 2025b,a; Team, 2025a; Contributors, 2025; Qiao et al., 2025; Shi et al., 2025c,b; Zhang et al., 2025a; Perincherry et al., 2025; Zhang et al., 2025b). VLN tasks are typically evaluated in photorealistic simulators e.g., Matterport3D, Habitat, with corresponding benchmarks, e.g., GOAT, HM3D-OVON, R2R (Khanna et al., 2024; Yokoyama et al., 2024; Puig et al., 2023). To train a VLN model with navigation abilities, many researchers employ imitation learning (IL) to mimic expert demonstrations (Ehsani et al., 2024) or reinforcement learning (RL) to optimize navigation policies through trial-and-error interactions with the environment (Zhu et al., 2017; Kulhánek et al., 2019; Cimurs et al., 2021). Our *Embodied-Reasoner* differs significantly from previous work. First, our model not only can navigate in unfamiliar environments but also possesses object interaction capabilities, such as opening drawers, picking up key chains inside, and transporting them to other locations. This object interaction capability enables the model to achieve enhanced object search and environmental exploration capabilities. Furthermore, unlike employing IL or RL to learn single skill, we introduce large reasoning models and language chain-of-thought to explicitly enhance its embodied reasoning capabilities for more complex human instructions and long-horizon interactive tasks.

System prompt:

```
You are a robot in given room. You need to complete the tasks according to human
instructions. We provide an Available_Actions set and the corresponding explanations
 for each action. Each step, you should select one action from Available_Actions.
```

Initialization prompt:

```
<image>This is an image from your frontal perspective. Please select an action from
the Available_Actions and fill in the arguments.
Task: {taskname}
Available_Actions: {{
"navigate to <object>": Move to the object.
"pickup <object>": Pick up the object.
"put in <object>": Put the item in your hand into or on the object.
"toggle <object>": Switch the object on or off.
"open <object>": Open the object (container), and you will see inside the object.
"close <object>": Close the object.
"observe": You can obtain image of your directly rear, left, and right perspectives.
"move forward": Move forward to see more clearly.
"end": If you think you have completed the task, please output "end".}}
Before making each decision, you can think, plan, and even reflect step by step, and
 then output your final action.
Your final action must strictly follow format: <DecisionMaking>Your Action</
DecisionMaking>, for example, <DecisionMaking>observe</DecisionMaking>.
```

Interaction prompt:

```
After executing your previous {action} , you get this new image above.
To complete your task, you can think step by step at first and then output your new
action from the Available_Actions.
Your action must strictly follow format: <DecisionMaking>Your Action</DecisionMaking
>, for example, <DecisionMaking>observe</DecisionMaking>.
```

Interaction feedback prompt 1:

```
<|feedback|>Action: {action} is illegal, {object} is the most relevant item in this
room and {action}. Object: {object} is not currently navigable, you can try "navigate
to <object>" to reach nearby, larger objects for closer observation.
```

Interaction feedback prompt 2:

```
<|feedback|>Action: {object} is illegal, Object: {object} is currently unavailable for
 interaction. Possible situations include: {object} does not exist in your current
view; you are too far away from {object}; the {object} cannot perform operation {
action}.\nYou can try "move forward" to approach the target object or "navigate to <
object>" to reach nearby, larger objects for closer inspection.
```

Interaction feedback prompt 3:

```
<|feedback|>Action: {action} is illegal, the name of the navigated object doesn't
quite match the object in the image, please try navigating to another object first.
```

Interaction feedback prompt 4:

```
<|feedback|>Action: {action} is illegal, the name of the object doesn't quite match
the object in the image, Please try interacting with another object or navigating to
 another object.
```

Figure C14: Detailed prompts for evaluation framework