Ego-embodied Reasoner: Egocentric Embodied Reasoning and Planning with MLLM via Reinforcement Learning

Zixuan Wang[*,1] **Danqi Zhao**[*,2] **Zhuo Cao**[*,4] **Puzhen Yuan**[*,3] **Chenxiao Yang**[*,4]
[1]Department of Physics, Tsinghua University
[2]School of Vehicle and Mobility, Tsinghua University
[3]Xingjian College, Tsinghua University
[4]Institute for Interdisciplinary Information Sciences, Tsinghua University
{wang-zx23, caozhuo23, cx-yang23}@mails.tsinghua.edu.cn

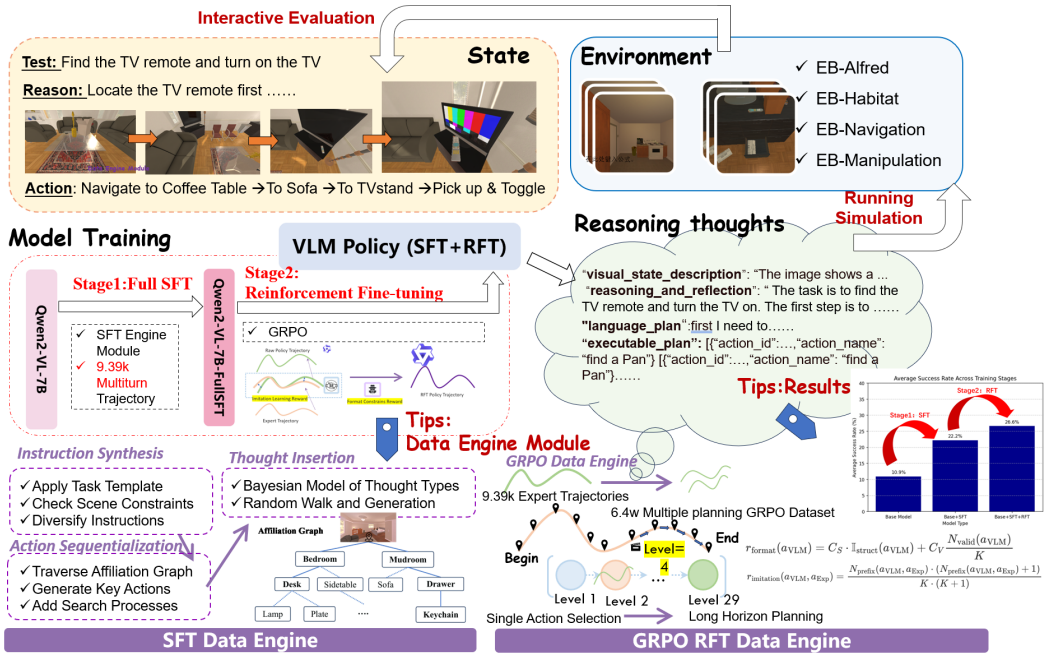

Figure 1: Overview of the EgoReasoner Framework.

Abstract

Current Multimodal Large Language Models (MLLMs) exhibit limitations in embodied reasoning, particularly in tasks requiring long-horizon planning, temporal logic, and spatial understanding. To address these challenges, we introduce the Ego-embodied Reasoner (EgoReasoner), a framework that enhances the planning capabilities of MLLMs through a multi-stage training pipeline. First, we synthesize a dataset of expert trajectories using a hybrid of rule-based logic and Large Language Model (LLM) calls. This data is then used for Supervised Fine-Tuning (SFT) to instill foundational planning and action-formatting abilities in the agent. To further refine its decision-making, we employ Group Relative Policy Optimization

(GRPO), a reinforcement learning technique that optimizes the policy by leveraging relative preferences between groups of sampled trajectories. We evaluate our method on interactive embodied search tasks from the EmbodiedBench suite. Our results show a significant performance gain, with the success rate improving from a 10.9% zero-shot baseline to 22.2% after SFT, and reaching 26.6% after GRPO refinement. This demonstrates the effectiveness of our approach in transforming smaller-scale MLLMs into competent agents for long-horizon embodied reasoning.

1 Introduction

Multimodal Large Language Models (MLLMs) have demonstrated impressive abilities in language grounding and visual understanding, making them attractive candidates for embodied reasoning and decision-making [1, 2]. However, when deployed in long-horizon embodied tasks, existing MLLMs still face significant limitations: they struggle with temporal reasoning, spatial understanding, and common-sense planning across multi-step decisions [3, 4]. This motivates our project to explore how to effectively enhance such capabilities within a computationally efficient framework.

We propose a lightweight yet effective pipeline for embodied agents built on MLLMs, aiming to improve decision-making through structured supervision and reinforcement learning. Our hybrid training framework begins by synthesizing high-quality expert interaction trajectories using a combination of rule-based logic and LLM-generated actions [5, 6]. These demonstrations are used for Supervised Fine-Tuning (SFT) on a visual-language model, instilling structured behavioral priors.

To further enhance long-horizon reasoning [7, 8], we utilize a reinforcement learning strategy called Group Relative Policy Optimization (GRPO) [9]. GRPO enhances policy learning by leveraging relative comparisons among groups of trajectory segments, facilitating more stable and effective optimization. This framework is validated on interactive embodied search tasks that require the agent to reason over space, time, and past actions. Our key insight is that even relatively small-scale MLLMs can be transformed into competent embodied agents through the combination of high-quality synthetic data, supervised pre-training, and reinforcement-based fine-tuning.

2 Related Work

2.1 Deep Thinking and Reasoning Models

The emergence of deep-thinking models such as OpenAI o1, DeepSeek-R1, and related systems has revolutionized complex reasoning tasks, particularly in mathematical and coding domains. These models employ a "slow-thinking" paradigm where they generate extensive reasoning chains before producing final answers, achieved through sophisticated reinforcement learning training on reasoning trajectories. However, extending this paradigm to embodied scenarios presents unique challenges. Unlike mathematical reasoning that relies primarily on logical deduction, embodied tasks require spatial understanding, temporal reasoning across interaction histories, and continuous adaptation to environmental feedback. Recent work by Zhao et al. [2] introduces Embodied-R, which addresses computational constraints by decoupling perception and reasoning through a collaborative framework combining large-scale vision-language models for perception with smaller language models for reasoning, trained via reinforcement learning with novel logical consistency rewards.

2.2 Tool Use and Interactive Clarification

A critical aspect of embodied reasoning involves effective tool use and handling of ambiguous or incomplete user intents. Traditional tool learning frameworks assume explicit and unambiguous queries, which diverges from real-world scenarios where users often provide incomplete or imprecise instructions. Zhang et al. [10] address this challenge through ASKTOACT, a self-correcting clarification framework that leverages the structural mapping between queries and tool invocation solutions. Their approach generates high-quality training data by systematically removing key parameters from complete queries while retaining them as ground truth, enabling automated construction of clarification dialogues. The framework incorporates error-correction mechanisms that allow models to

detect and recover from mistakes during multi-turn interactions, achieving significant improvements in intent recovery and clarification efficiency.

2.3 Embodied Spatial Reasoning

Spatial reasoning represents a fundamental challenge in embodied AI, requiring models to perceive and reason about spatial relationships from sequential visual observations. Current approaches often struggle with the complexity of spatio-temporal relationships in video data and the distinct characteristics of egocentric visual observations. The Embodied-R framework [2] tackles these challenges through a collaborative architecture that separates perception and reasoning components, enabling efficient processing of continuous visual streams while maintaining computational tractability. Their approach demonstrates that small-scale language models can achieve comparable performance to much larger multimodal reasoning models when properly trained with reinforcement learning techniques that emphasize logical consistency.

2.4 Training Paradigms for Embodied Agents

The development of effective embodied agents requires sophisticated training paradigms that can handle the complexity of interactive environments. Recent approaches have explored various combinations of supervised fine-tuning (SFT) and reinforcement learning (RL) to enhance agent capabilities. Zhang et al. [1] present a comprehensive three-stage training pipeline that progressively builds agent capabilities through imitation learning, self-exploration via rejection sampling, and self-correction through reflection tuning. This iterative approach addresses the challenge of learning from both expert demonstrations and self-generated experience, enabling models to develop robust reasoning patterns for long-horizon tasks.

Our work builds upon these foundations by proposing a lightweight yet effective pipeline that combines high-quality synthetic data generation with a hybrid training framework incorporating both SFT and GRPO. This approach addresses the computational efficiency concerns raised by previous work while maintaining the sophisticated reasoning capabilities demonstrated by deep-thinking models, specifically tailored for long-horizon embodied interactive tasks.

3 Methods

We formulate the problem as a sequential decision-making task. At each timestep t, the agent receives a visual-language observation $o_t \in \mathcal{O}$ and chooses an action $a_t \in \mathcal{A}$ according to its policy π_θ, which is instantiated as an MLLM. The interaction history is $h_t = \{o_0, a_0, \ldots, o_t\}$. Given a language-grounded goal g and a task prompt L, a full trajectory is $\tau = \{g, o_0, a_0, \ldots, o_n, a_n\}$. Our goal is to train π_θ to generate trajectories that successfully complete the task. The overall framework is depicted in Figure 1.

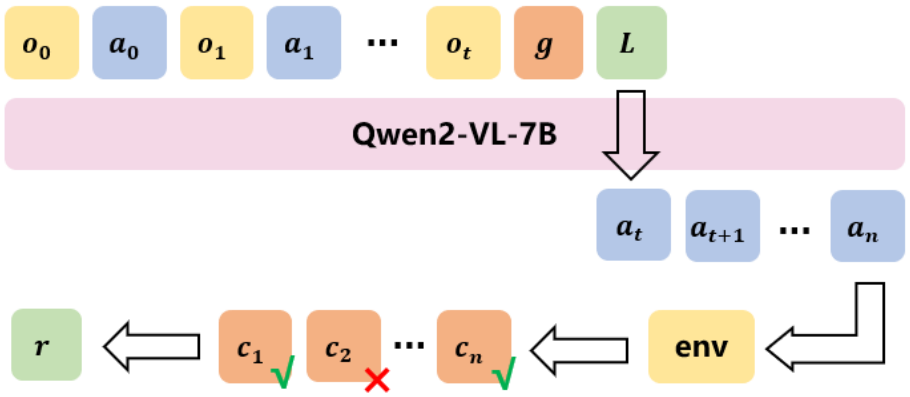

Figure 2: Overview of Model Architecture

The main idea of EgoReasoner method is to finetune an MLLM (e.g. Qwen-VL in practice) leveraging both imitation learning (SFT) and reinforcement learning (GRPO),

3.1 Trajectory Synthesis and Supervised Fine-Tuning (SFT)

To bootstrap the training process, we first construct a dataset of expert-like demonstrations using a hybrid system. This system combines deterministic, rule-based logic for common sub-tasks with LLM calls to generate semantically rich and diverse action sequences. This approach allows us to generate high-quality training data without requiring expensive human annotation.

Specifically, the synthesis process contains three main steps:

- **Instruction Synthesis:** We pre-define four types of task templates including search, manipulation, transportation and composition. Each time we apply the sampled template with also sampled objects and then check scene consistencies and constraints that determine whether the whole description is valid and tractable. To enhance the diveristy of instructions, we also exploit LLMs to modify the styles and difficulties of generated instructions.
- **Action Sequentialization:** Given the scene (in simulator), an affiliation graph can be built based on the co-relations of scene objects. By traversing the affiliation graph under the guidance of instruction can we generate the key actions in sequence. [1, 2]
- **Thought Insertion:** Prior works in psychology have shown the bayesian model of five types of human thoughts: analysis, planning, reflection, spatial reasoning and verification. [11] We use Monte Carlo sampling to randomly walk on the probabilistic graph and generate thoughts via LLMs.

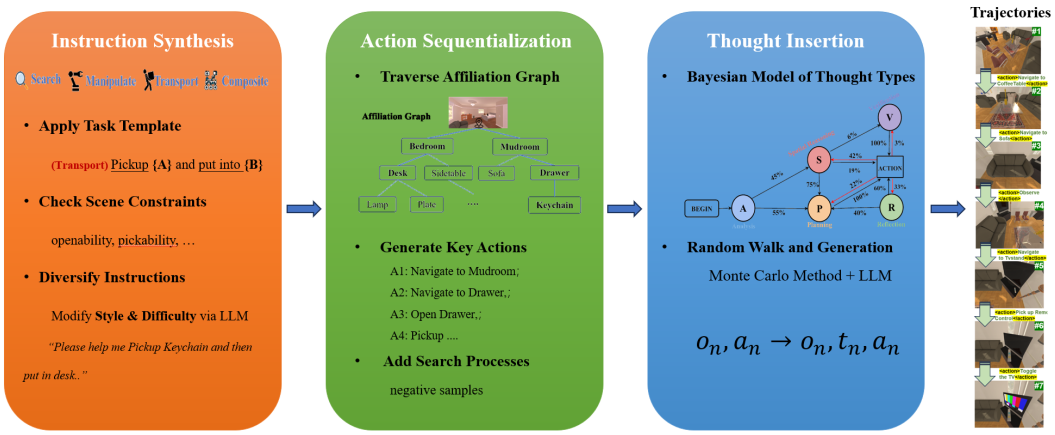

Figure 3: SFT Data Engine

Using these synthesized trajectories, we have collected ∼10K trajectories and can perform SFT on our base model, Qwen2-VL-Instruct-7B. The objective of this stage is to teach the model structured reasoning patterns and the correct action format (e.g., a `<Think>...<\Think><Act>...<\Act>` structure). This pre-training serves as a crucial initialization for the subsequent reinforcement learning phase. We utilize the Swift framework for the SFT process.

3.2 Refinement via Group Relative Policy Optimization (GRPO)

Following SFT, we further enhance the agent's reasoning abilities using GRPO. GRPO is a preference-based RL algorithm that operates by comparing groups of trajectories. For a given task prompt, we generate multiple candidate trajectories using the current policy. These trajectories are then evaluated and ranked, and the relative preferences are used to update the policy. This group-wise comparison provides a more robust and stable learning signal than pairwise comparisons or absolute rewards. We explore two variants for generating rewards and preferences. [9]

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$
$$\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{\min\left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}\hat{A}_{i,t}, \text{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}, 1-\varepsilon, 1+\varepsilon\right)\hat{A}_{i,t}\right] - \beta\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right]\right\},$$

Figure 4: GRPO Object

3.2.1 GRPO with Imitation Learning (Offline)

In this offline paradigm, we leverage the expert trajectories from our synthesized dataset. The reward function is designed to align the agent's behavior with the expert demonstrations. For a reference action sequence of length k and a generated prefix that matches the reference for n steps, we define a multi-step accuracy reward:

$$R_{\text{accuracy}} = \frac{n(n+1)}{k(k+1)}$$

This reward encourages the model to not only select the correct first action but also to produce longer, correct sequences. An additional format reward penalizes outputs that deviate from the required thinking and action template. The final reward is a weighted sum of these components. This approach allows us to fine-tune the model without requiring continuous interaction with a live simulator.

3.2.2 GRPO with Online Simulation

To enable the agent to learn from its own mistakes and explore novel strategies, we also implement an online version of GRPO. In this setup, for each training step, the agent's generated action sequence is executed in a live simulator instance. The reward is then computed based on the outcome of this execution. The total reward R_{sim} is a composite signal:

- **Format Reward**: Penalizes incorrectly formatted outputs, as in the offline version.
- **Task Completion Reward**: A large positive reward is given for successfully completing the task, with a small penalty for failure.
- **Length Penalty**: A minor penalty is applied to overly long action sequences to encourage efficiency.
- **Repetition Penalty**: Repeated, oscillatory behaviors (e.g., moving back and forth) are penalized to promote more effective exploration.

4 Experiments and Results

4.1 Experimental Setup

Benchmarks We conduct our evaluations on selected subsets of the EmbodiedBench platform [12], a comprehensive benchmark for embodied vision-language agents. We focus on two representative subsets:

- **EB-ALFRED**: Emphasizes multi-step interactive task completion with natural language instructions in realistic household environments.
- **EB-Habitat**: Leverages the Habitat simulator for complex indoor navigation and interaction tasks.

These benchmarks provide a balanced evaluation of an agent's reasoning, planning, and perception skills in visually rich, interactive settings.

Baselines We compare our EgoReasoner-7B model against several strong baseline MLLMs, including Qwen2-VL-7B-Instruct, Qwen2-VL-72B-Instruct, Qwen-VL-Max, and the Embodied-Reasoner-7B model [**embodied-reasoner**].

4.2 Performance Evaluation

First, we evaluated the stage-wise improvement of our training pipeline. As shown in Table 1, our 7B model starts with a zero-shot success rate of 10.9%. After SFT, the success rate more than doubles to 22.2

Table 1: Stage-wise Performance Improvement of EgoReasoner-7B. Success Rate (%) is reported on a held-out set of embodied search tasks.

Model Stage	Success Rate (%)
Zero-shot (Base Model)	10.9
After SFT	22.2
After GRPO (Final Model)	26.6

Next, we compare our final model with other state-of-the-art MLLMs on the EB-ALFRED and EB-Habitat benchmarks. As shown in Table 2, our EgoReasoner-7B significantly outperforms other models of similar size (Qwen2-VL-7B and Embodied-Reasoner-7B) and achieves performance competitive with, and in some cases exceeding, much larger models like Qwen2-VL-72B. Specifically, on EB-ALFRED, our model achieves a 40.1% success rate, surpassing the Embodied-Reasoner-7B by 1.6 percentage points. These results highlight our framework's ability to instill powerful reasoning capabilities into a relatively small model. A qualitative example of our agent's reasoning process is shown in Figure 5.

Table 2: Model Success Rate (%) on EB-ALFRED and EB-Habitat Benchmarks.

Model	EB-ALFRED Success (%)	EB-Habitat Success (%)
Qwen2-VL-7B-Ins	1.7	18.3
Embodied-Reasoner-7B [**embodied-reasoner**]	38.5	41.1
EgoReasoner-7B (Ours)	**40.1**	**41.5**
Qwen2-VL-72B-Ins	33.7	35.7
Qwen-VL-Max (235B)	41.3	45.3

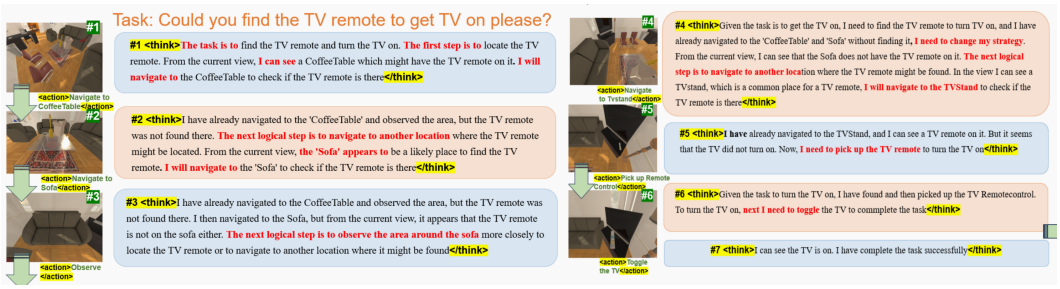

Figure 5: Visualization of EgoReasoner's Interactive Embodied Reasoning Process. The agent receives a visual observation, thinks through its next steps, and executes an action towards completing the high-level goal.

5 Conclusion and Discussion

In this work, we presented EgoReasoner, a framework demonstrating that a combination of Supervised Fine-Tuning (SFT) on high-quality synthetic data and subsequent refinement via Group Relative Policy Optimization (GRPO) can significantly elevate the high-level reasoning and planning capabilities of smaller-scale Vision-Language Models (VLMs) for long-horizon embodied tasks. The strong performance of our 7B model, which rivals and even surpasses much larger baseline models, underscores the potential of this methodology for creating efficient yet powerful embodied

agents. However, while celebrating these promising results, it is imperative to critically examine the capability boundaries of current VLMs as high-level planners and to reflect on the path toward truly general-purpose embodied intelligence.

5.1 On the Emergence and Mechanism of VLM-based Planning

The success of EgoReasoner is largely attributable to its staged training strategy, which systematically builds agent capabilities. The initial SFT stage is critical; by using "expert trajectories," it instills a structured *behavioral prior* into the model. This process constrains the model's vast output space, transforming it from a general-purpose language generator into a goal-oriented policy network that understands the fundamental syntax of the task, such as the `<Think>...<\Think><Act>...<\Act>` chain-of-thought format. This provides a high-quality initialization, preventing the subsequent reinforcement learning phase from exploring an intractably large and inefficient search space.

The GRPO stage is where the model's decision-making and reasoning abilities are truly refined. Unlike traditional RL algorithms such as PPO, which rely on absolute and often noisy reward signals, preference-based methods like GRPO offer a more stable and effective learning signal. By optimizing the policy based on the relative ranking of groups of trajectories, GRPO excels in complex domains like embodied AI, where designing a dense and perfectly-calibrated reward function is notoriously difficult. The integration of online simulation further allows the agent to learn directly from the consequences of its own mistakes, not merely by imitating an expert. This fosters the development of more robust and generalizable strategies, explaining the significant performance boost observed after the GRPO fine-tuning phase.

5.2 Capability Boundaries and Inherent Limitations

Despite the encouraging outcomes, we must acknowledge the profound limitations that define the current frontier of VLM-based planners.

Lack of Physical Commonsense and Causal Reasoning. The "reasoning" exhibited by the current model is fundamentally a high-dimensional statistical pattern matching, not a deep, causal understanding of the physical world. For instance, the model learns not to move through a wall because such an action is absent from successful trajectories in its training data, not because it comprehends the physical concept of solidity. Its knowledge is correlational, not causal. Consequently, if faced with a novel scenario involving unfamiliar physics (e.g., a permeable holographic barrier), the model would likely fail, as its knowledge base lacks the grounding in first principles required for true generalization. It effectively knows *what* to do in familiar contexts, but not *why* it works.

Challenges in Open-World Generalization. Our model has demonstrated proficiency on the EB-ALFRED and EB-Habitat benchmarks. However, these environments, while complex, remain fundamentally closed-world systems with finite sets of objects, layouts, and affordances. The agent's performance is intrinsically tied to the quality of the SFT data and the constraints of its predefined, discrete action space. Its ability to generalize to a truly open-world environment—with novel objects, unforeseen challenges, and different interaction dynamics—remains a significant open question. A crucial next step for the field is to move beyond fixed action sets toward enabling agents to perform *skill discovery*, autonomously learning new capabilities from interaction.

Training Efficiency and the Sim-to-Real Gap. The online GRPO training paradigm, while effective, is computationally exorbitant. Generating and evaluating multiple trajectories for each training step requires massive-scale parallel simulation, posing a significant bottleneck to scalability. Furthermore, this research was conducted entirely in simulation. The *sim-to-real gap*—the discrepancy between a simulator and the complexities of the real world, including sensor noise, actuation errors, and unpredictable dynamics—remains a formidable and unaddressed challenge. Success in Habitat does not guarantee robust performance for a physical robot, and our current framework does not explicitly tackle this crucial transfer problem.

5.3 Reflections and Future Directions

Based on this analysis, we propose that future research in this domain should focus on the following key directions.

Building Explicit and Interactable World Models. The field must move beyond purely implicit world models. Future work could explore training VLMs to explicitly predict the physical consequences of their actions, for instance, by forecasting future video frames or changes in a geometric state representation. Integrating VLMs with symbolic structures, such as knowledge graphs, could also pave the way for more robust causal reasoning and planning.

Towards Lifelong Learning and Continual Adaptation. The current SFT+GRPO pipeline represents an offline training paradigm. True autonomy requires agents capable of *lifelong learning*, continuously updating their knowledge and policies from an ongoing stream of experience post-deployment. Integrating mechanisms for online and continual learning is essential for developing agents that can adapt to changing environments over extended periods.

In conclusion, EgoReasoner charts an effective course for imbuing embodied agents with the reasoning capabilities of VLMs through a careful synthesis of supervised learning and reinforcement learning. The journey ahead, however, requires moving beyond simulated benchmarks to directly confront the challenges of physical causality, open-world generalization, and operational safety. Only then can we hope to realize the ultimate vision of general-purpose embodied intelligence that can safely and effectively operate alongside humans in the real world.

A Appendix

A.1 Predefined Action Space

The agent interacts with the environment using a discrete, predefined action space. This ensures that the model's outputs can be directly mapped to executable commands in the simulator. The action space includes navigation commands (e.g., `MoveAhead`, `RotateLeft`), interaction commands (e.g., `PickupObject`, `SliceObject`), and state-changing commands (e.g., `OpenObject`, `CloseObject`). A visualization of the key actions is provided in Figure 6.

```
{
    "navigate to <object>\": Move to the object.
    "pickup <object>\": Pick up the object.
    "put <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.\n\"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\".
}.
```

Figure 6: A subset of the predefined action space available to the agent.

A.2 GRPO Training Data Example

Figure 7 shows an example of the data format used for GRPO training. Each item consists of a prompt (including visual and textual context) and multiple sampled responses from the policy, along with their corresponding rewards. This structure is essential for the group-wise comparison at the core of the GRPO algorithm.

```
{
    "messages":
    [
        {
         "role": "system",
         "content": "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."},
        {
         "role": "user",
         "content": "<image>This is an image from your initial frontal perspective.
Please select an action from the Available_Actions and fill in the arguments.\nTask:
\"Get a Newspaper from the room.\"\nAvailable_Actions: <Predefined Action set>\nYour
final action must strictly follow format: <DecisionMaking>Your Action</DecisionMaking>,
for example, <DecisionMaking>observe</DecisionMaking>.Before making each decision, you
can think, plan, and even reflect step by step, and then output your final
action.\n\nPrevious actions taken: navigate to CoffeeTable, navigate to CoffeeTable,
navigate to SideTable, navigate to DiningTable"},
     ],
    "action": ["pickup Newspaper", "navigate to Sofa", "pickup Newspaper"],
    "images":
["./data/images/navigate1pickup0/FloorPlan228_single_pickup_5798_fix/FloorPlan228_1_init.
png"]
}
```


Figure 7: Example of a data instance for GRPO training.

References

[1] W. Zhang et al. "Embodied-Reasoner: Synergizing Visual Search, Reasoning, and Action for Embodied Interactive Tasks". In: *arXiv preprint arXiv:2503.21696* (2025). URL: https://arxiv.org/abs/2503.21696.

[2] B. Zhao et al. "Embodied-R: Collaborative Framework for Activating Embodied Spatial Reasoning in Foundation Models via Reinforcement Learning". In: *arXiv preprint arXiv:2504.12680* (2025). URL: https://arxiv.org/abs/2504.12680.

[3] G. Wang, Y. Xie, Y. Jiang, et al. "Voyager: An Open-Ended Embodied Agent with Large Language Models". In: *arXiv preprint arXiv:2305.16291* (2023). URL: https://arxiv.org/abs/2305.16291.

[4] M. Ahn, A. Brohan, N. Brown, et al. "Do as I Can, Not as I Say: Grounding Language in Robotic Affordances". In: *arXiv preprint arXiv:2204.01691* (2022). URL: https://arxiv.org/abs/2204.01691.

[5] W. Yuan, J. Duan, V. Blukis, et al. "RoboPoint: A Vision-Language Model for Spatial Affordance Prediction for Robotics". In: *arXiv preprint arXiv:2406.10721* (2024). URL: https://arxiv.org/abs/2406.10721.

[6] R. Ramrakhya, M. Chang, X. Puig, et al. "Grounding Multimodal LLMs to Embodied Agents that Ask for Help with Reinforcement Learning". In: *arXiv preprint arXiv:2504.00907* (2025). URL: https://arxiv.org/abs/2504.00907.

[7] J. Bai et al. "Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond". In: *arXiv preprint arXiv:2308.12966* (2023). URL: https://arxiv.org/abs/2308.12966.

[8] J. Yang et al. "Thinking in Space: How Multimodal Large Language Models See, Remember, and Recall Spaces". In: *arXiv preprint arXiv:2412.14171* (2024). URL: https://arxiv.org/abs/2412.14171.

[9] DeepSeek-AI et al. "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning". In: *arXiv preprint arXiv:2501.12948* (2025). URL: https://arxiv.org/abs/2501.12948.

[10] Xuan Zhang et al. "Asktoact: Enhancing llms tool use via self-correcting clarification". In: *arXiv preprint arXiv:2503.01940* (2025).

[11] Yizhong Wang et al. *Self-Instruct: Aligning Language Models with Self-Generated Instructions*. 2023. arXiv: 2212.10560 [cs.CL]. URL: https://arxiv.org/abs/2212.10560.

[12] R. Yang, H. Chen, J. Zhang, et al. "EmbodiedBench: Comprehensive Benchmarking Multi-modal Large Language Models for Vision-Driven Embodied Agents". In: *arXiv preprint arXiv:2502.09560* (2025). URL: https://arxiv.org/abs/2502.09560.