VIKI-R: Coordinating Embodied Multi-Agent Cooperation via Reinforcement Learning

Li Kang[1,2*], **Xiufeng Song**[1,2*], **Heng Zhou**[3,2*], **Yiran Qin**[2,5†], **Jie Yang**[5], **Xiaohong Liu**[1], **Philip Torr**[4], **Lei Bai**[2†], **Zhenfei Yin**[4†]
[1]Shanghai Jiao Tong University [2]Shanghai Artificial Intelligence Laboratory
[3]University of Science and Technology of China [4]University of Oxford
[5]The Chinese University of Hong Kong, Shenzhen
{faceong02, sparklexfantasy, hengzzzhou}@gmail.com
* Equal contribution † Corresponding author
https://faceong.github.io/VIKI-R/

Figure 1: Embodied multi-agent cooperation involves two key aspects: (1) cross-embodiment collaboration, where different embodiments are required for different tasks (e.g., washing requires a humanoid, while only wheeled robots can fetch from high cabinets); and (2) efficient coordination, where agents work in parallel (e.g., multiple arms passing apples while a humanoid washes them) to improve overall efficiency. To support such fine-grained teamwork, we propose *VIKI-Bench*, which structures the process into three levels of visual reasoning: Level 1 – agent activation, Level 2 – task planning, and Level 3 – trajectory perception, aiming to realize an embodied multi-agent system.

Abstract

Coordinating multiple embodied agents in dynamic environments remains a core challenge in artificial intelligence, requiring both perception-driven reasoning and scalable cooperation strategies. While recent works have leveraged large language models (LLMs) for multi-agent planning, a few have begun to explore vision-language models (VLMs) for visual reasoning. However, these VLM-based approaches remain limited in their support for diverse embodiment types. In this work, we introduce *VIKI-Bench*, the first hierarchical benchmark tailored for embodied multi-agent cooperation, featuring three structured levels: agent activation, task planning, and trajectory perception. *VIKI-Bench* includes diverse robot

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

embodiments, multi-view visual observations, and structured supervision signals to evaluate reasoning grounded in visual inputs. To demonstrate the utility of *VIKI-Bench*, we propose *VIKI-R*, a two-stage framework that fine-tunes a pretrained vision-language model (VLM) using Chain-of-Thought annotated demonstrations, followed by reinforcement learning under multi-level reward signals. Our extensive experiments show that *VIKI-R* significantly outperforms baselines method across all task levels. Furthermore, we show that reinforcement learning enables the emergence of compositional cooperation patterns among heterogeneous agents. Together, *VIKI-Bench* and *VIKI-R* offer a unified testbed and method for advancing multi-agent, visual-driven cooperation in embodied AI systems.

1 Introduction

In the science-fiction film *I, Robot* [49], the super-computer VIKI orchestrates thousands of NS-5 robots, illustrating the extraordinary coordination capabilities of heterogeneous robotic agents. This fictional depiction highlights a fundamental challenge in artificial intelligence: enabling multiple embodied agents to collaborate in dynamic, real-world environments. As illustrated in Fig. 1, addressing this challenge is critical for advancing multi-agent systems capable of achieving effective, large-scale coordination: (1) Real-world tasks often necessitate specialized embodiments—for instance, reaching high cabinets may call for a robot with extended reach, while delicate tasks demand manipulators with fine-grained control. (2) Cooperative behaviors substantially enhance task efficiency through parallelization and mutual assistance.

Recent advances have demonstrated the potential of large language models (LLMs) in enabling multi-agent planning [6, 8, 58]. While these LLM-based approaches have made significant progress in high-level coordination, only a few works have explored the use of vision-language models (VLMs) for perception-driven reasoning [27, 47, 59]. However, existing VLM-based methods remain limited by the lack of embodiment diversity. As a result, the ability to reason about visual observations in heterogeneous multi-agent settings remains an underexplored challenge.

To address these gaps, we introduce ***VIKI-Bench***, a comprehensive benchmark for evaluating collaborative capabilities in embodied multi-agent systems. As illustrated in Fig. 1, *VIKI-Bench* is designed around three levels of task: Agent Activation, Task Planning, and Trajectory Perception. Each task provides multi-view visual input and incorporates a diverse set of heterogeneous robots. Moreover, *VIKI-Bench* provides a multi-dimensional evaluation framework that assesses execution feasibility, task completion and planning efficiency. To the best of our knowledge, *VIKI-Bench* is the first comprehensive benchmark specifically designed to evaluate the reasoning capabilities of VLMs in hierarchical embodied multi-agent cooperation.

To advance reasoning capabilities in the multi-agent system, we introduce ***VIKI-R***, a VLM-based framework that fosters reasoning abilities in multi-agent cooperation. Inspired by [13, 28, 43], our approach first grounds a pretrained VLM in task understanding through Chain-of-Thought annotations, then optimizes it via Reinforcement Learning, leveraging hierachical supervision in *VIKI-Bench*. Extensive experimental results demonstrate that *VIKI-R* significantly outperforms baseline methods across all three task levels, highlighting the effectiveness of the proposed approach.

In summary, the main contributions of this paper are as follows:

◇ We introduce *VIKI-Bench*, the first hierarchical benchmark for embodied multi-agent cooperation, which consists of three structured task levels: agent activation, high-level task planning, and low-level trajectory perception. The benchmark features heterogeneous robot types, multi-view visual inputs, and structured supervision signals to enable comprehensive evaluation.

◇ We propose *VIKI-R*, a two-stage learning framework that enhances visual reasoning capabilities in embodied multi-agent systems by using hierarchical reward signals to learn structured reasoning across diverse tasks, enabling generalizable cooperation in complex environments.

◇ Extensive experimental results demonstrates the effectiveness of *VIKI-R* in *VIKI-Bench*. Our analysis highlights the importance of hierarchical supervision and reveals how reinforcement learning facilitates the emergence of compositional collaboration patterns in embodied environments.

Table 1: **Comparison to similar embodied benchmarks.** We compare VIKI-Bench to embodied AI benchmarks, focusing on natural language and multi-agent collaboration tasks. [Keys: **Views:** EGO (Ego-centric view), GL (Global view). **H.E.:** Coordination among Heterogeneous Embodiments.]

	Environment	Language	Visual	Views	H.E.	Tasks Num
Overcooked [7]	2D	✓		-		4
RoCo [30]	3D	✓		-		6
WAH [35]	3D	✓		-		1,211
Co-ELA [58]	3D	✓		-		44
FurnMove [19]	3D	✓	✓	EGO		30
PARTNR [8]	3D	✓		-	✓	100,000
RoboCasa [31]	3D	✓	✓	EGO	✓	100
LLaMAR (MAP-THOR) [32]	3D	✓	✓	EGO, GL	✓	225
VIKI-Bench (Ours)	3D	✓	✓	EGO, GL	✓	23,737

2 Related Work

Embodied Multi-Agent Cooperation Real-world embodied tasks often require cooperation among multiple agents. Existing studies [2, 14, 24, 36, 39, 57, 66, 38, 64, 68, 67] have explored this problem in various application domains. Research focuses on multi-agent task allocation [25, 33, 46] and joint decision-making [45, 58]. A significant body of recent work [6, 15, 22, 31, 40, 62, 69] leverages large language models (LLMs) to handle high-level reasoning and planning. Some recent works leverage video generation models [5, 37, 52, 53, 54, 55] to construct multi-agent world models [59], achieving promising results on specific tasks. However, these approaches lack visual grounding, limiting their ability to reason about spatial constraints and perceptual affordances. While a few recent efforts [47, 63, 65] incorporate vision-language models (VLMs) to obtain a more grounded understanding of the environment, research on heterogeneous multi-agent cooperation remains sparse—particularly in settings requiring fine-grained visual reasoning and embodied perception. In contrast, our work incorporates both agent heterogeneity and visual reasoning to support complex, perception-driven collaboration.

Visual Reasoning Visual reasoning requires vision-language models (VLMs) to interpret and reason over visual observations to perform complex tasks. It has been applied in areas such as geometric problem-solving [12, 42, 60], robotic [16, 17, 20, 65] and scientific research [21, 29, 61]. Previous work has explored enhancing visual reasoning in VLMs through multi-stage supervision. For example, LLaVA-CoT [50] applies multi-stage supervised fine-tuning (SFT) with chain-of-thought [48] prompting. With the introduction of a rule-based reinforcement learning (RL) method, DeepSeek-R1 [13] demonstrates significant improvements in reasoning performance. Recent works [26, 28, 43] incorporate RL to further enhance visual reasoning capabilities. Our work shows that R1-style methods perform better in multi-agent embodied visual reasoning tasks.

Embodied multi-agent benchmarks Recent research [1, 7, 58, 8, 31] has developed several embodied multi-agent benchmarks to evaluate collaborative behaviors. In 2D environments, LLM-Co [1] and Overcooked [7] study coordination in game play, but the simplified 2D settings limit their abilities in physical interaction. For 3D environments, a thread of work has focused on language-guided cooperative planning for embodied tasks. For instance, WAH [35] examines social intelligence in household scenarios. PARTNR [8] evaluates visual planning and reasoning under LLM-based evaluation. Other benchmarks target multi-agent manipulation. RocoBench [30] conducts object interaction tasks within a tabletop environment. FurnMove [19] requires collaboration on synchronized furniture arrangement. LLaMAR [32] focuses on multi-agent task planning and refinement, with 225 task instances, emphasizing task planning and verification. VIKI-Bench goes a step further by offering a broader evaluation framework, featuring over 23,000 task instances across 100 diverse scenes and multiple robot types that bridges both planning and manipulation domains.

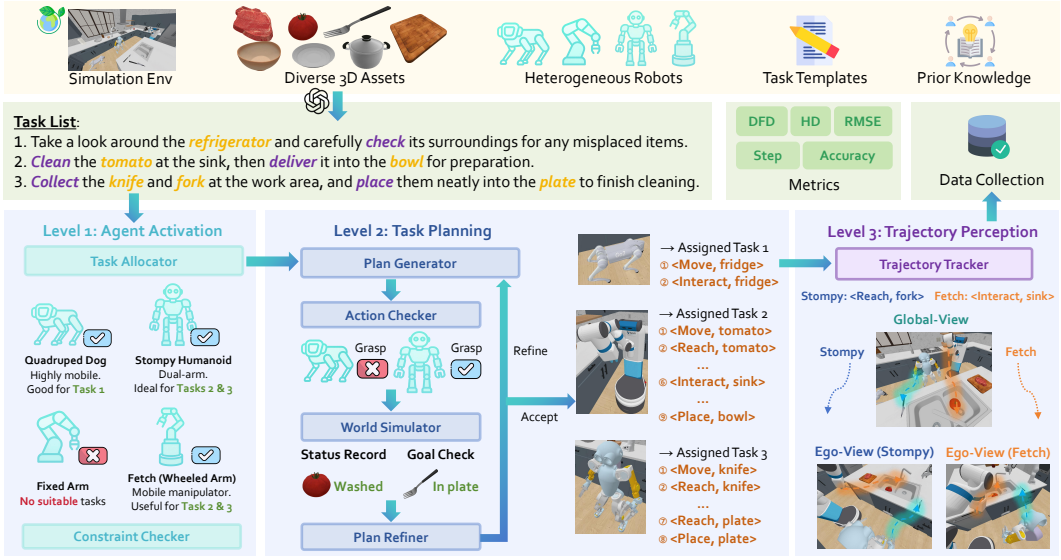


Figure 2: Overview of *VIKI-Bench*. *VIKI-Bench* is a hierarchical benchmark for evaluation on multi-agent embodied cooperation, featuring visual reasoning tasks in three levels: (1) Agent Activation, where robots are selected based on the scene image and the task context; (2) Task Planning, where a structured multi-agent action plan is generated, verified, and refined; and (3) Trajectory Perception, where the fine-grained motion trajectory of each agent is tracked from egocentric views. The benchmark involves diverse robot types and complex 3D environments, with multiple metrics for quantitative evaluation.

3 VIKI-Bench

3.1 Overview

We introduce ***VIKI-Bench***, a hierarchical benchmark for studying visual reasoning in embodied multi-agent collaboration, as illustrated in Fig. 2. *VIKI-Bench* covers three levels of tasks: (1) Agent Activation, which selects appropriate agents to activate by considering the task description and the scene image; (2) Task Planning, which requires generating an ordered sequence of action primitives of multiple agents; and (3) Trajectory Perception, which involves predicting the motion trajectories of all agents. Each task includes a language instruction, with global visual observations provided for the first two levels, and egocentric views used for the trajectory perception level. Spanning thousands of tasks across heterogeneous robot morphologies and diverse household-to-industrial layouts, *VIKI-Bench* offers a concise yet comprehensive benchmark for scalable multi-agent cooperation.

3.2 Data Generation

3.2.1 Agent Activation

We formulate the agent activation task as a visual reasoning problem, where the task allocator selects a set of appropriate robots among all agents to complete the task. Each sample is formatted as an instruction-question pair, consisting of an image observation O and a task instruction I. The expected answer is a set of selected agents $R = \{r_j\}, j \in [1, M]$ chosen from the visible agent pool $\mathcal{A}_{\text{visible}}$ based on embodiment reasoning and task affordance.

To generate ground truth labels, we construct task-specific templates that specify which agent types are required or not required for solving the task, given the task goal and environmental context. These templates are grounded in embodiment rules and capability-based constraints (*e.g.*, mobile agents for navigation, dual-arm agents for bimanual manipulation).

To encourage interpretable reasoning, we adopt a chain-of-thought format in which the model is expected to: (1) analyze the task requirements, (2) visually identify the robots present, (3) assess each robot's suitability, and (4) conclude the final selection. For data generation, we employ GPT-4o [34]

as the task allocator g_{act}, prompting it with the task template and the corresponding image context. The activation result is then obtained as $R = g_{\text{act}}(I, O)$. A verification module C_{act} is used to automatically check whether the generated labels conform to embodiment-grounded task constraints, followed by human inspection to correct failure cases and ensure overall label quality.

3.2.2 Task Planning

We construct task planning data as question-answer pairs according to the environment and specific instructions. To describe high-level operations of agents in the environment, we design basic primitive set P (*e.g.*, move, grasp, etc.) as the atomic operations of all agents. The planning answer is designed as a sequence of action descriptions $A = \{a_1, a_2, ...a_N\}$, where N is the length of the sequence. Each action description is formed as $a_i = (r_i, t_i, p_i, d_i)$, where r_i, t_i, p_i, d_i denotes the agent, the timestep, the primitive and the destination of action a_i, respectively.

To generate effective planning in versatile environments, we use GPT-4o as the plan generator g_{plan} and introduce an iterative refinement process. Given an instruction I, the corresponding observation O, and the primitive set P, the generator first decomposes the instruction into a set of goals G, and generates a possible planning result A_0, as $A_0 = g_{plan}(I, O, P)$. Then, an Action Checker C verifies the feasibility of each action based on the rules of primitives, followed by a World Simulator S recording the position and status of interactive entities in the environment. Subsequently, a Plan Refiner R checks the completion of the goals. For any failure in planning, the refiner provides detailed feedback as an additional instruction, which is concatenated with the original instruction for the generator to revise the planning result until success. This procedure is formulated as follows.

Algorithm 1 Iterative Refinement Process

Require: Plan Generator g_{plan}, Instruction I_0, Goals G, Observation O, Primitives P
Ensure: Successful Planning A

$Success \leftarrow False$
$I \leftarrow I_0$
while $\neg Success$ **do**
  $A \leftarrow g_{plan}(I, O, P)$
  $Act\_success \leftarrow C(act), \forall act \in A$ ▷ Action feasibility check
  $Status \leftarrow S(A)$
  $Goal\_success \leftarrow is\_successful(Status, goal), \forall goal \in G$ ▷ Goal check
  $Success \leftarrow Act\_success \wedge Goal\_success$
  $I \leftarrow I + R(Act\_success, Goal\_success)$ ▷ Update feedback instruction
end while

3.2.3 Trajectory Perception

We formulate trajectory perception in multi-agent environments as a spatial keypoint prediction problem, where the model predicts motion trajectories from egocentric observations based on the task instruction. Unlike prior work [8, 20] that focuses solely on the observing agent, our setting requires predicting both the trajectory of the ego agent and those of other visible agents to facilitate collaboration, which are referred as the *ego-trajectory* and *partner-trajectories*, respectively. Given an egocentric RGB image I and an action description $a_i = (r_i, t_i, p_i, d_i)$ indicating the ongoing execution, the model predicts a set of 2D trajectories $\mathcal{L} = \{T_k\}, k \in [1, M]$, where M is the number of agents in the scene, and $L_k = \{(x_j, y_j)\}_{j=1}^{L}$ denotes a temporally ordered spatial motion for agent r_k in coordinate sequences.

To construct these samples, we sample diverse egocentric observations from simulated multi-agent scenes with the corresponding task descriptions. Based on the egocentric observations and detailed instructions for each visible agent, the trajectory of each agent is manually annotated by formulating feasible a motion path in the form of coordinate sequences. All data undergoes human verification to ensure temporal consistency and spatial alignment with the instruction and environment.

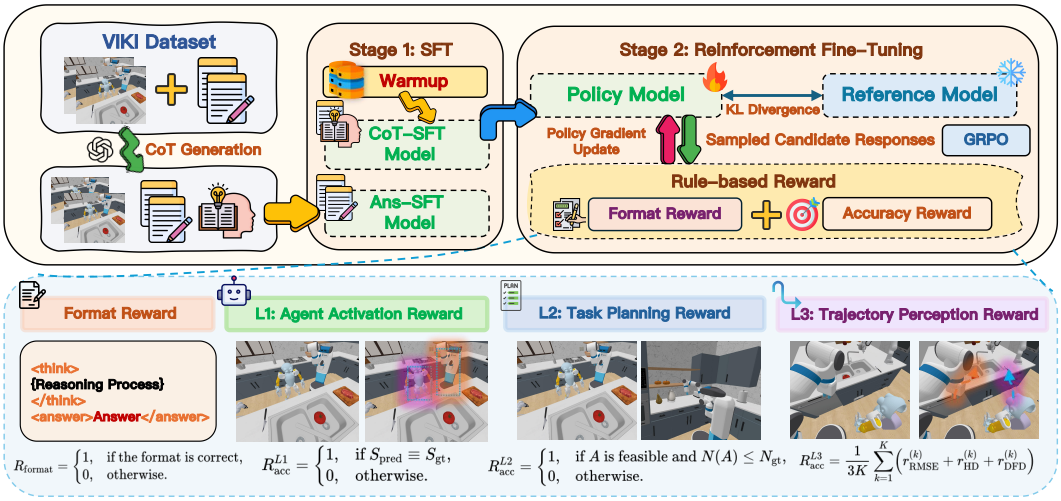


Figure 3: Framework of *VIKI-R*. We adopted supervised fine-tuning (SFT) and reinforcement fine-tuning on the VIKI dataset, incorporating format and accuracy rewards to optimize the policy model.

3.3 Data Statistics

The VIKI benchmark comprises over 20,000 multi-agent task samples across 100 diverse scenes derived from the RoboCasa [31] based on ManiSkill3 [44], each with fine-grained object configurations and varied spatial layouts. The dataset involves 6 types of heterogeneous embodied agents (e.g., humanoids, wheeled arms, quadrupeds) interacting with over 1,000 unique asset combinations. Each scene provides both global and egocentric camera views to support perception and planning. More details are provided in Appendix C.1.

4 VIKI-R

4.1 Overview

We introduce *VIKI-R*, a two-stage fine-tuning framework that endows vision–language models with robust visual reasoning abilities, as shown in Fig. 3. In the first stage, *SFT-based Warmup*, the model undergoes supervised fine-tuning on high-quality Chain-of-Thought (CoT) annotations, optimizing the likelihood of both intermediate reasoning steps and final answers. This stage instructs the model to acquire domain-specific reasoning patterns. In the second stage, *Reinforcement Fine-Tuning*, the policy is refined using the Grouped Relative Proximal Optimization (GRPO) algorithm [41]. For each visual–question pair, grouped candidate answers are sampled and evaluated using a composite reward function based on answer format and correctness. Standardized advantages are then computed to guide policy updates under a KL-divergence constraint, ensuring stable and consistent improvement.

4.2 Training Objectives

SFT-based Warmup In the first phase, we employ Supervised Fine-Tuning (SFT) with data annotated with Chain-of-Thought (CoT) reasoning process. Each training instance is denoted as (x, q, r, a), where x represents the visual input, q the associated task, r the intermediate reasoning steps, and a the final answer. The SFT objective maximizes the joint likelihood of the reasoning and answer tokens conditioned on the input:

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{(x,q,r,a)\sim\mathcal{D}} \sum_{t=1}^{T} \log \pi_\theta \left(y_t \mid x, q, y_{<t} \right), \tag{1}$$

where $\mathcal{D}$ is the CoT-annotated dataset, $y = [r, a]$ is the concatenated sequence of reasoning and answer tokens, and π_θ denotes the model's token distribution.

Reinforcement Fine-Tuning Starting from the Chain-of-Thought initialized policy π_{CoT}, we perform group-relative reinforcement fine-tuning following the GRPO formulation. Given an input $s = (x, q)$, we sample a group of G candidate outputs $\{a_i\}_{i=1}^{G}$, each receiving a reward R_i. We compute the empirical mean $\bar{R}$ and standard deviation σ_R of the group, and define the standardized advantage for each candidate as

$$\mathcal{A}_i = \frac{R_i - \bar{R}}{\sigma_R}. \tag{2}$$

This group-relative normalization highlights candidates that outperform their peers and stabilizes learning by removing dependence on absolute reward scale.The probability ratio between the current policy π_θ and the reference policy π_0 (initialized from π_{CoT}) is defined as

$$r_i(\theta) = \frac{\pi_\theta(a_i \mid s)}{\pi_0(a_i \mid s)}. \tag{3}$$

The clipped surrogate objective is then given by

$$\mathcal{L}^{\mathrm{CLIP}}(\theta) = \mathbb{E}_s\left[\sum_{i=1}^{G} \min\Big(r_i(\theta)\mathcal{A}_i,\ \mathrm{clip}\big(r_i(\theta),\, 1-\epsilon,\, 1+\epsilon\big)\mathcal{A}_i\Big)\right], \tag{4}$$

where ϵ is the clipping coefficient that bounds the policy update step.

Finally, the policy is optimized under a KL-divergence constraint to maintain proximity to the reference policy:

$$\mathcal{J}(\theta) = \mathcal{L}^{\mathrm{CLIP}}(\theta) - \beta\, \mathrm{D}_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid s) \,\|\, \pi_0(\cdot \mid s)\big), \tag{5}$$

where $\beta > 0$ controls the trust region enforced by the KL regularizer. This GRPO formulation enables stable and sample-efficient reinforcement fine-tuning, allowing the policy to leverage group-relative advantages for compositional spatial reasoning.

4.3 Reward Design

To guide the model towards both structured output and task accuracy, we formulate the overall reward into a *format reward* and a task-specific *accuracy reward*, as:

$$R = \lambda_1 \times R_{\mathrm{format}} + \lambda_2 \times R_{\mathrm{acc}}, \tag{6}$$

where R_{format} enforces the output format and R_{acc} corresponds to the three subtask rewards, as defined below. λ_1 and λ_2 refer to the weights of both rewards, respectively.

Format Reward To encourage explicit reasoning, we assign a binary format reward: the model receives 1 point if it correctly encloses the intermediate reasoning steps within `<think>...</think>` and the final answer within `<answer>...</answer>`, and 0 otherwise. By enforcing these tags, we prompt the model to articulate its chain-of-thought before delivering the answer, thereby improving interpretability and guiding systematic reasoning.

Agent Activation Reward We define the agent activation reward as an exact-match indicator between the predicted agent set S_{pred} and the ground-truth set S_{gt}:

$$R_{\mathrm{acc}}^{L1} = \begin{cases} 1, & \text{if } S_{\mathrm{pred}} \equiv S_{\mathrm{gt}}, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Task Planning Reward While multiple feasible plans may exist, we define the reward to favor efficient solutions. Specifically, a predicted plan A only receives the reward if it is feasible and its length does not exceed that of the ground-truth plan N_{gt}. Let $N(A)$ denote the length of the predicted action sequence A, the task planning reward is defined as:

$$R_{\mathrm{acc}}^{L2} = \begin{cases} 1, & \text{if } A \text{ is feasible and } N(A) \le N_{\mathrm{gt}}, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Details on how plan feasibility is checked are provided in Appendix C.2.

Trajectory Perception Reward Let $P^{(k)} = \{p_t^{(k)}\}_{t=1}^{T}$ and $G^{(k)} = \{g_t^{(k)}\}_{t=1}^{T}$ denote the predicted and ground-truth trajectories for agent k, respectively. To evaluate trajectory prediction quality for each agent k, we compute three normalized standard geometric distance metrics between the predicted trajectory and the ground-truth trajectory: Root Mean Square Error (RMSE, denoted as $\hat{d}_{\mathrm{RMSE}}$), Hausdorff Distance (HD, denoted as $\hat{d}_{\mathrm{HD}}$)[18], and Discrete Fréchet Distance (DFD, denoted as $\hat{d}_{\mathrm{DFD}}$)[11]. Since smaller distances indicate better alignment between predicted and ground-truth trajectories, we transform the distance $\hat{d}$ into a reward-like score using the transformation $r = 1 - \hat{d}$. The final trajectory perception reward is defined as:

$$R_{\mathrm{acc}}^{L3} = \frac{1}{3K} \sum_{k=1}^{K} \left(r_{\mathrm{RMSE}}^{(k)} + r_{\mathrm{HD}}^{(k)} + r_{\mathrm{DFD}}^{(k)} \right), \tag{9}$$

5 Experiments

Table 2: Performance comparison across the three hierarchical task levels of VIKI-Bench. Best scores are highlighted in **bold**, and the second-best scores are underlined.

Method	VIKI-L1	VIKI-L2			VIKI-L3			
	$\mathrm{ACC_{ID}} \uparrow$	$\mathrm{ACC_{ID}} \uparrow$	$\mathrm{ACC_{OOD}} \uparrow$	$\mathrm{ACC_{AVG}} \uparrow$	RMSE ↓	HD ↓	DFD ↓	AVG ↓
Closed-Source Models								
GPT-4o	18.40	22.56	10.02	17.50	100.80	115.34	131.05	115.73
Claude-3.7-Sonnet	12.40	19.44	0.57	11.82	283.31	323.53	346.88	317.91
Gemini-2.5-Flash-preview	31.40	20.00	10.51	16.17	453.89	519.14	540.80	504.61
Open-Source Models								
Qwen2.5-VL-72B-Instruct	11.31	8.40	1.20	5.49	81.31	94.62	113.15	96.36
Qwen2.5-VL-32B-Instruct	9.50	3.60	0.00	2.15	88.48	99.80	119.78	102.69
Llama-3.2-11B-Vision	0.40	0.50	0.00	0.30	192.69	223.57	231.85	216.04
Qwen2.5VL-3B-Instruct								
Zero-Shot	1.95	0.22	0.00	0.13	96.22	114.93	130.98	114.04
+Ans SFT	35.29	81.06	30.71	60.74	74.70	90.28	102.26	89.08
+VIKI-R-Zero	20.40	0.00	0.00	0.00	80.36	95.36	120.27	98.66
+VIKI-R	74.10	93.61	32.11	68.78	75.69	90.25	103.65	89.86
Qwen2.5VL-7B-Instruct								
Zero-Shot	4.26	0.44	0.00	0.26	81.93	103.82	112.91	99.55
+Ans SFT	72.20	**96.89**	25.62	68.13	65.32	81.20	90.89	79.14
+VIKI-R-Zero	**93.59**	0.17	0.00	0.10	67.42	85.30	95.32	82.68
+VIKI-R	93.00	95.22	**33.25**	**69.25**	**64.87**	**79.23**	**89.36**	**77.82**

5.1 Experimental Setup

Training Paradigms and Baselines To assess the impact of different training strategies on performance and generalization, we compare the following methods: (1)Ans-SFT: a supervised fine-tuning (SFT) approach focusing solely on answer generation. (2)VIKI-R-Zero: a reinforcement learning (RL) variant that applies GRPO directly, without any prior CoT activation. (3)VIKI-R: our two-phase scheme—first SFT on a small CoT-annotated subset, followed by GRPO-based RL. All variants use Qwen2.5-VL-Instruct [4] as the base model in both 3B and 7B sizes to study the effect of model scale. For a comprehensive comparison, we include open-source models[4, 23]and leading closed-source systems GPT-4o [34], gemini-2.5-flash-preview [9]) and claude-3.7-sonnet[3] as baselines. Detailed hyperparameters and additional setup information are provided in Appendix D.

Evaluation Metrics We adopted task-specific metrics to evaluate performance across the three stages of the *VIKI-Bench*. For agent activation (VIKI-L1), we report classification accuracy based on whether the selected agents match the ground truth. For task planning (VIKI-L2), we evaluate accuracy based on whether the predicted plan is both feasible and no longer than the ground-truth plan, reflecting correctness and execution efficiency. For trajectory perception (VIKI-L3), we evaluate the predicted trajectories using RMSE, Hausdorff Distance (HD) [18] and Discrete Fréchet Distance (DFD) [11], which measure spatial and temporal alignment with ground-truth motion paths.

5.2 Overall Performance Analysis

Tab. 2 highlights three main observations. First, when comparing open-source and closed-source models under zero-shot evaluation (without any *VIKI-Bench* training), closed-source models hold a clear advantage. Among closed-source systems, Gemini-2.5-Flash-preview achieves the highest agent activation accuracy, while GPT-4o excels at trajectory perception. In contrast, both Gemini and Claude exhibit almost no trajectory-prediction capability. Second, the model scale critically affects open-source VLM performance. The 72B-parameter Qwen2.5-VL matches or even surpasses some closed-source baselines on perception metrics, but reducing the model to 32B parameters incurs substantial drops in both planning accuracy and trajectory quality. This underscores the importance of model capacity for handling complex multi-agent visual reasoning. Third, our two-stage fine-tuning framework *VIKI-R* outperforms purely supervised Ans-SFT and VIKI-R-zero. While Ans-SFT yields strong in-domain improvements, it fails to generalize to out-of-domain scenarios. These results confirm that integrating reinforcement learning substantially enhances visual reasoning capabilities in hierarchical multi-agent cooperation.

5.3 Feedback-Driven Iterative Refinement

We compare two planning strategies: standard sampling (up to k attempts without guidance) and feedback-driven sampling (injecting feedback between attempts). Tab. 3 demonstrates the impact of feedback-driven sampling. By injecting feedback between failed attempts, GPT-4o achieves improvements of 1.9% at `pass@3` and 3.6% at `pass@6`. Claude-3-7-Sonnet sees gains of 1.5% and 2.3% and Gemini-2.5-Flash records increases of 1.8% and 3.0%. On average, feedback-driven sampling boosts `pass@3` by 1.7% and `pass@6` by 3.0%, highlighting that iterative feedback effectively steers the model away from repeated mistakes and yields more reliable plans.

Table 3: Task planning success rates (%) under two sampling strategies. `pass@k` denotes the probability of obtaining at least one valid plan within k independent attempts, while `pass@k_fb` is measured when feedback is appended after each failed attempt.

Model	`pass@1`	`pass@3`	`pass@3_fb`	`pass@6`	`pass@6_fb`
GPT-4o [34]	18.4	18.7	20.6	18.7	22.3
Claude-3.7-Sonnet [3]	12.4	12.4	13.9	12.5	14.8
Gemini-2.5-Flash-preview [9]	31.4	31.6	33.4	31.7	34.7

5.4 Ablation Study

Tab. 4 demonstrates the impact of step penalty. By incorporating a constraint-based penalty, VIKI-R achieves improvements by 39.7% and 88.0% in the accuracy of out-of-domain and in-domain tasks, respectively. These results underscore the effectiveness of the step penalty in generalization and execution accuracy. Besides, the steps of action length is reduced by an average of 1.92 steps, highlighting the critical role of penalizing unnecessary steps to enforce concise planning. Overall, the step penalty promotes more transferable and efficient planning strategies.

Table 4: Effect of the step penalty on 1,000 challenging reasoning tasks sampled from both the out-of-domain (OOD-H) and in-domain (ID-H) splits. Δ Steps measures the average difference between the action length of predicted plan and the ground-truth plan.

Variant	$\mathrm{ACC}_{\text{OOD-H}} \uparrow$	$\mathrm{ACC}_{\text{ID-H}} \uparrow$	$\Delta\text{Steps} \downarrow$
VIKI-R (with step penalty)	46.8	96.0	0.05
VIKI-R (without step penalty)	7.1	8.0	1.97

5.5 Real-World Case Study

To validate our approach beyond simulation, we perform real-world evaluations on **VIKI-L1** tasks using a dual-arm RealMan robot with an Agilex mobile base and two Franka Research 3 arms. We

instantiate 100 tasks with GPT-4o [34]–generated instructions and evaluate leading vision–language models in a zero-shot manner.

Table 5: Real-world accuracy on VIKI-L1 tasks (%).

Model	Acc.	Model	Acc.
Qwen2.5-VL-72B	14.0	Qwen2.5-VL-32B	8.0
Claude-3.7-Sonnet	15.0	Gemini-2.5-Flash	**28.0**

These experiments confirm that while real-world execution remains challenging, current VLMs already demonstrate basic embodied reasoning and scene grounding. Bridging this gap between simulation and reality offers a promising direction for future large-scale embodied benchmarks.

5.6 Insights from Training

Throughout our experiments, we identified several key behaviors that illustrate both the strengths and limitations of GRPO in our hierarchical multi-agent setting.

Dependence on Base Policy Quality The effectiveness of GRPO depends critically on the competence of the pretrained policy. In VIKI-L2 planning, the zero-shot model produces almost no valid plans, and VIKI-R-Zero yields negligible improvement. By contrast, in the VIKI-L1 activation and VIKI-L3 perception tasks where the base policy already generates some correct responses—GRPO delivers clear performance gains. These observations indicate that reinforcement-based fine-tuning requires an initial set of correct rollouts to guide effective policy updates.

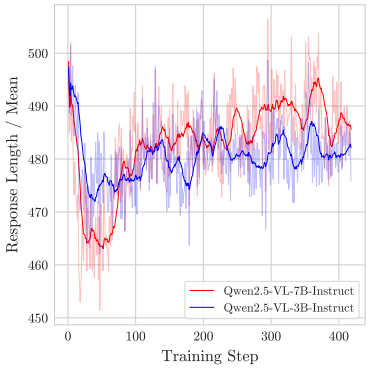


Figure 4: Response length of the Qwen2.5-VL-3B/7B-Instruct model at training time.

Evolution of Response Length We tracked the average token length of model outputs during VIKI-R training in Fig. 4. In the early stages, output length decreases as the model prioritizes format compliance to secure the format reward. Once format accuracy saturates, the policy shifts focus toward maximizing task correctness, and output length gradually increases to include the necessary reasoning details.

5.7 Evaluation with Human Experts

We further compare VIKI-R with human experts and analyze its potential for human–agent collaboration. As shown in Table 6, VIKI-R achieves high success rates on VIKI-L1 and VIKI-L2 (98% and 96.5%), and demonstrates competitive reasoning consistency under the more challenging VIKI-L3 setting, with notable reductions in RMSE, HD, and DFD metrics compared to human performance.

Table 6: Human expert and model comparison on VIKI-Bench. Higher is better for VIKI-L1/L2, and lower is better for VIKI-L3 metrics.

Task / Method	VIKI-L1 (%)	VIKI-L2 (%)	VIKI-L3 (RMSE)	VIKI-L3 (HD)	VIKI-L3 (DFD)
Human Expert	98.00	96.50	48.84	58.97	71.30
VIKI-R (Ours)	93.00	69.25	64.87	79.23	89.36

6 Conclusion

This paper presents *VIKI-Bench*, a hierarchical benchmark for evaluating vision-language models in embodied multi-agent collaboration. We further introduce *VIKI-R*, a two-stage framework that combines supervised pretraining and reinforcement learning to solve multi-agent tasks across activation, planning, and perception levels. While our study focuses on simulated environments, extending this framework to real-world settings and dynamic agents remains promising future work.

Acknowledgment

This work was supported by the Shanghai Municipal Science and Technology Major Project.

References

[1] Saaket Agashe, Yue Fan, Anthony Reyna, and Xin Eric Wang. Llm-coordination: evaluating and analyzing multi-agent coordination abilities in large language models. *arXiv preprint arXiv:2310.03903*, 2023.

[2] Xing An, Celimuge Wu, Yangfei Lin, Min Lin, Tsutomu Yoshinaga, and Yusheng Ji. Multi-robot systems and cooperative object transport: Communications, platforms, and challenges. *IEEE Open Journal of the Computer Society*, 4:23–36, 2023.

[3] AI Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 2024.

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

[5] Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. *arXiv preprint arXiv:2412.03572*, 2024.

[6] Xiaohe Bo, Zeyu Zhang, Quanyu Dai, Xueyang Feng, Lei Wang, Rui Li, Xu Chen, and Ji-Rong Wen. Reflective multi-agent collaboration based on large language models. *Advances in Neural Information Processing Systems*, 37:138595–138631, 2024.

[7] Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems*, 32, 2019.

[8] Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Mottaghi, Priyam Parashar, et al. Partnr: A benchmark for planning and reasoning in embodied multi-agent tasks. *arXiv preprint arXiv:2411.00081*, 2024.

[9] Google DeepMind. Gemini 2.5. `https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025`, 2025.

[10] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*, 2023.

[11] Thomas Eiter and Heikki Mannila. Computing discrete fréchet distance. Technical Report CD-TR 94/64, Christian Doppler Laboratory for Expert Systems, 1994.

[12] Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, et al. G-llava: Solving geometric problem with multi-modal large language model. *arXiv preprint arXiv:2312.11370*, 2023.

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

[14] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*, 2024.

[15] Xudong Guo, Kaixuan Huang, Jiale Liu, Wenhui Fan, Natalia Vélez, Qingyun Wu, Huazheng Wang, Thomas L Griffiths, and Mengdi Wang. Embodied llm agents learn to cooperate in organized teams. *arXiv preprint arXiv:2403.12482*, 2024.

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

[17] Yingdong Hu, Fanqi Lin, Tong Zhang, Li Yi, and Yang Gao. Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning. *arXiv preprint arXiv:2311.17842*, 2023.

[18] Daniel P Huttenlocher, Gregory A Klanderman, and William J Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

[19] Unnat Jain, Luca Weihs, Eric Kolve, Ali Farhadi, Svetlana Lazebnik, Aniruddha Kembhavi, and Alexander Schwing. A cordial sync: Going beyond marginal policies for multi-agent embodied tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 471–490. Springer, 2020.

[20] Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. *arXiv preprint arXiv:2502.21257*, 2025.

[21] Yanhao Jia, Xinyi Wu, Qinglin Zhang, Yiran Qin, Luwei Xiao, and Shuai Zhao. Towards robust evaluation of stem education: Leveraging mllms in project-based learning. *arXiv preprint arXiv:2505.17050*, 2025.

[22] Shyam Sundar Kannan, Vishnunandan LN Venkatesh, and Byung-Cheol Min. 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, 2024.

[23] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[24] Rui Li, Zixuan Hu, Wenxi Qu, Jinouwen Zhang, Zhenfei Yin, Sha Zhang, Xuantuo Huang, Hanqing Wang, Tai Wang, Jiangmiao Pang, et al. Labutopia: High-fidelity simulation and hierarchical benchmark for scientific embodied agents. *arXiv preprint arXiv:2505.22634*, 2025.

[25] Jiaqi Liu, Chengkai Xu, Peng Hang, Jian Sun, Mingyu Ding, Wei Zhan, and Masayoshi Tomizuka. Language-driven policy distillation for cooperative driving in multi-agent reinforcement learning. *IEEE Robotics and Automation Letters*, 2025.

[26] Xiangyan Liu, Jinjie Ni, Zijian Wu, Chao Du, Longxu Dou, Haonan Wang, Tianyu Pang, and Michael Qizhe Shieh. Noisyrollout: Reinforcing visual reasoning with data augmentation. *arXiv preprint arXiv:2504.13055*, 2025.

[27] Xinzhu Liu, Di Guo, Huaping Liu, and Fuchun Sun. Multi-agent embodied visual semantic navigation with scene prior knowledge. *IEEE Robotics and Automation Letters*, 7(2):3154–3161, 2022.

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

[29] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.

[30] Zhao Mandi, Shreeya Jain, and Shuran Song. Roco: Dialectic multi-robot collaboration with large language models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 286–299. IEEE, 2024.

[31] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

[32] Sid Nayak, Adelmo Morrison Orozco, Marina Have, Jackson Zhang, Vittal Thirumalai, Darren Chen, Aditya Kapoor, Eric Robinson, Karthik Gopalakrishnan, James Harrison, et al. Long-horizon planning for multi-agent robots in partially observable environments. *Advances in Neural Information Processing Systems*, 37:67929–67967, 2024.

[33] Kazuma Obata, Tatsuya Aoki, Takato Horii, Tadahiro Taniguchi, and Takayuki Nagai. Lip-llm: Integrating linear programming and dependency graph with large language models for multi-robot task planning. *IEEE Robotics and Automation Letters*, 2024.

[34] OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya

Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. Gpt-4o system card, 2024.

[35] Xavier Puig, Tianmin Shu, Shuang Li, Zilin Wang, Yuan-Hong Liao, Joshua B Tenenbaum, Sanja Fidler, and Antonio Torralba. Watch-and-help: A challenge for social perception and human-ai collaboration. *arXiv preprint arXiv:2010.09890*, 2020.

[36] Yiran Qin, Li Kang, Xiufeng Song, Zhenfei Yin, Xiaohong Liu, Xihui Liu, Ruimao Zhang, and Lei Bai. Robofactory: Exploring embodied agent collaboration with compositional constraints. *arXiv preprint*

arXiv:2503.16408, 2025.

[37] Yiran Qin, Zhelun Shi, Jiwen Yu, Xijun Wang, Enshen Zhou, Lijun Li, Zhenfei Yin, Xihui Liu, Lu Sheng, Jing Shao, et al. Worldsimbench: Towards video generation models as world simulators. *arXiv preprint arXiv:2410.18072*, 2024.

[38] Yiran Qin, Ao Sun, Yuze Hong, Benyou Wang, and Ruimao Zhang. Navigatediff: Visual predictors are zero-shot navigation assistants. *arXiv preprint arXiv:2502.13894*, 2025.

[39] Yiran Qin, Enshen Zhou, Qichang Liu, Zhenfei Yin, Lu Sheng, Ruimao Zhang, Yu Qiao, and Jing Shao. Mp5: A multi-modal open-ended embodied system in minecraft via active perception. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16307–16316. IEEE, 2024.

[40] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

[41] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[42] Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See-Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. Math-llava: Bootstrapping mathematical reasoning for multimodal large language models. *arXiv preprint arXiv:2406.17294*, 2024.

[43] Huajie Tan, Yuheng Ji, Xiaoshuai Hao, Minglan Lin, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. Reason-rft: Reinforcement fine-tuning for visual reasoning. *arXiv preprint arXiv:2503.20752*, 2025.

[44] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, et al. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2024.

[45] Weizheng Wang, Ike Obi, and Byung-Cheol Min. Multi-agent llm actor-critic framework for social robot navigation. *arXiv preprint arXiv:2503.09758*, 2025.

[46] Yongdong Wang, Runze Xiao, Jun Younes Louhi Kasahara, Ryosuke Yajima, Keiji Nagatani, Atsushi Yamashita, and Hajime Asama. Dart-llm: Dependency-aware multi-robot task decomposition and execution using large language models. *arXiv preprint arXiv:2411.09022*, 2024.

[47] Yujin Wang, Quanfeng Liu, Zhengxin Jiang, Tianyi Wang, Junfeng Jiao, Hongqing Chu, Bingzhao Gao, and Hong Chen. Rad: Retrieval-augmented decision-making of meta-actions with vision-language models in autonomous driving. *arXiv preprint arXiv:2503.13861*, 2025.

[48] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[49] Wikipedia. I, robot (film). `https://en.wikipedia.org/wiki/I,_Robot_(film)`, 2004.

[50] Guowei Xu, Peng Jin, Li Hao, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step, 2024. *URL https://arxiv. org/abs/2411.10440*.

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

[52] Jiwen Yu, Jianhong Bai, Yiran Qin, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Context as memory: Scene-consistent interactive long video generation with memory retrieval. *arXiv preprint arXiv:2506.03141*, 2025.

[53] Jiwen Yu, Yiran Qin, Haoxuan Che, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, Kun Gai, Hao Chen, and Xihui Liu. A survey of interactive generative video. *arXiv preprint arXiv:2504.21853*, 2025.

[54] Jiwen Yu, Yiran Qin, Haoxuan Che, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Position: Interactive generative video as next-generation game engine. *arXiv preprint arXiv:2503.17359*, 2025.

[55] Jiwen Yu, Yiran Qin, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Gamefactory: Creating new games with generative interactive videos. *arXiv preprint arXiv:2501.08325*, 2025.

[56] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[57] Hangtao Zhang, Chenyu Zhu, Xianlong Wang, Ziqi Zhou, Shengshan Hu, and Leo Yu Zhang. Badrobot: Jailbreaking llm-based embodied ai in the physical world. *arXiv preprint arXiv:2407.20242*, 2024.

[58] Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. *arXiv preprint arXiv:2307.02485*, 2023.

[59] Hongxin Zhang, Zeyuan Wang, Qiushi Lyu, Zheyuan Zhang, Sunli Chen, Tianmin Shu, Behzad Dariush, Kwonjoon Lee, Yilun Du, and Chuang Gan. Combo: compositional world models for embodied multi-agent cooperation. *arXiv preprint arXiv:2404.10775*, 2024.

[60] Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Ziyu Guo, Shicheng Li, Yichi Zhang, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Bin Wei, et al. Mathematical visual instruction tuning with an automatic data engine. *arXiv preprint arXiv:2407.08739*, 2024.

[61] Sha Zhang, Suorong Yang, Tong Xie, Xiangyuan Xue, Zixuan Hu, Rui Li, Wenxi Qu, Zhenfei Yin, Tianfan Fu, Di Hu, et al. Position: Intelligent science laboratory requires the integration of cognitive and embodied ai. *arXiv preprint arXiv:2506.19613*, 2025.

[62] Zirui Zhao, Wee Sun Lee, and David Hsu. Large language models as commonsense knowledge for large-scale task planning. *Advances in Neural Information Processing Systems*, 36:31967–31987, 2023.

[63] Sipeng Zheng, Jiazheng Liu, Yicheng Feng, and Zongqing Lu. Steve-eye: Equipping llm-based embodied agents with visual perception in open worlds. *arXiv preprint arXiv:2310.13255*, 2023.

[64] Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *European Conference on Computer Vision*, pages 55–72. Springer, 2025.

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

[66] Enshen Zhou, Yiran Qin, Zhenfei Yin, Yuzhou Huang, Ruimao Zhang, Lu Sheng, Yu Qiao, and Jing Shao. Minedreamer: Learning to follow instructions via chain-of-imagination for simulated-world control. *arXiv preprint arXiv:2403.12037*, 2024.

[67] Enshen Zhou, Qi Su, Cheng Chi, Zhizheng Zhang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, and He Wang. Code-as-monitor: Constraint-aware visual programming for reactive and proactive robotic failure detection. *arXiv preprint arXiv:2412.04455*, 2024.

[68] Heng Zhou, Hejia Geng, Xiangyuan Xue, Zhenfei Yin, and Lei Bai. Reso: A reward-driven self-organizing llm-based multi-agent system for reasoning tasks. *arXiv preprint arXiv:2503.02390*, 2025.

[69] Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, et al. Sotopia: Interactive evaluation for social intelligence in language agents. *arXiv preprint arXiv:2310.11667*, 2023.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitation section is provided in Supplementary Material Section A.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: We provide the full set of assumptions and complete, correct proofs for each theoretical result.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper fully discloses all necessary information required to reproduce the main experimental results relevant to its core claims and conclusions.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We specify all the training and test details in Supplementary Material Section D.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We provide the statistical information of the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We specify all the sufficient information on the computer resources in Supplementary Material Section D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We provide the broader impacts in Supplementary Material Section B.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: The paper isn't relevant with any data or models that have a high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We explicitly mentioned and properly respected the license and terms of assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We introduce new assets with well documentations.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: The paper is not relevant with human subjects.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

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

A Limitations

Although VIKI-Bench and VIKI-R advance embodied multi-agent cooperation, several challenges remain unresolved.

First, the hierarchical task levels proposed in VIKI-Bench, while useful for structuring cooperative interactions, may not fully reflect all dynamic conditions and cooperative practice in real-world scenarios. Real environments involve unforeseen obstacles, shifting objectives, and adaptive agent behaviors that are difficult to model within a fixed hierarchical framework.

Second, while VIKI-R's hierarchical reward designs effectively enhance agent performance, their effectiveness relies on multi-level fine-tuning, which raises the needs for computational costs. A possible solution is to devise a more precious reward structure that adapt to complex environmental variations without requiring excessive tuning.

Additionally, although the emergent collaborative behaviors show prominent performance in multi-level tasks, the reasoning process and interpretability remains underexplored. Such studies are crucial for ensuring trust and facilitating human-AI collaboration in real-world deployments.

In conclusion, while our benchmark and framework provide a strong foundation for multi-agent cooperation, they face challenges in adaptability, computational complexity and interpretability. Future work could focus on expanding task diversity, improving framework design, and enhancing model transparency. Addressing these limitations will be essential for developing more robust and scalable multi-agent systems.

B Broader Impacts

The development of VIKI-Bench and VIKI-R has significant influence on both embodied multi-agent research and real-world applications. By introducing a hierarchical benchmark for embodied multi-agent cooperation, this work enables multi-dimensional evaluation and comparison of vision-language models (VLMs) in complex, dynamic environments. In practice, VIKI-Bench can accelerate progress in real-world robotics applications, such as warehouse automation, autonomous driving, and collaborative industrial robots, where heterogeneous agents must coordinate under visual uncertainty.

The proposed VIKI-R framework demonstrates how fine-tuning VLMs with reasoning and reinforcement learning can improve multi-agent decision-making, potentially leading to more adaptable and efficient autonomous systems. However, the deployment of such systems raises important considerations, including safety, fairness in decision-making, and the potential relationships between human and robots. Future work should address these challenges while leveraging VIKI-Bench's structured supervision to ensure robustness and interpretability in real-world scenarios. Ultimately, this research paves the way for more sophisticated AI systems capable of seamless cooperation in visually rich, dynamic environments.

C Details of VIKI-Bench

This section introduces details of VIKI-Bench, including the overview of data statistics, plan feasibility checking and experimental setup.

C.1 Data Statistics

The datasets in *VIKI-Bench* are organized into three hierarchical levels:

- VIKI-L1: Agent activation (10,714 samples; 8,171 training, 2,043 testing).
- VIKI-L2: Symbolic planning (10,714 samples; 7,196 in-domain training, 1,800 in-domain testing, and 1,218 out-of-domain testing from held-out scenes).
- VIKI-L3: Trajectory perception (2,309 samples; 1,767 training, 442 testing).

This multi-stage structure facilitates comprehensive evaluation of high-level coordination and low-level motion prediction in realistic, dynamic environments.

C.2 Plan Feasibility Check

The VIKI-L2 task must produce a plan list that successfully passes the feasibility checker described in Section 3.2.2—identical to the process used during data generation.

D Implementation Details

As summarized in Table 7, during GRPO we operate under a PPO framework using the VLLM engine with the XFORMERS attention backend. Inputs are truncated beyond a 4096-token prompt length and a 2048-token response length. We employ a total batch size of 256, subdivided into PPO minibatches of 128 and micro-batches of 10 per GPU; the actor network is optimized with a learning rate of 1×10^{-6}. KL-divergence regularization (coefficient 0.01, `low_var_kl` variant) is used, while entropy regularization and KL-based rewards are disabled. Gradient checkpointing reduces memory footprint, targeting 60% memory utilization during rollout with five rollouts per prompt, and both chunked prefill and eager execution are disabled. Finally, we train VIKI-L1 for five epochs, VIKI-L2 for fifteen epochs, and VIKI-L3 for two epochs.

We build upon the `Qwen2.5-VL-3B` and `Qwen2.5-VL-7B` backbones, orchestrating the entire pipeline with the open-source `verl` framework and employing `LLamaFactory` for supervised fine-tuning. All experiments run on a single node equipped with eight NVIDIA A800 GPUs. Three tasks—VIKI-L1, VIKI-L2, and VIKI-L3—with the following data splits: **VIKI-L1**: 10714 samples (500 cold-start SFT, 8171 training, 2043 testing);**VIKI-L2**: 10174 samples, of which 1218 are held out as OOD and the remaining 8956 as ID (with 500 cold-start CoT, 7196 training, 1800 ID testing);**VIKI-L3**: 2309 samples (100 cold-start CoT, 1767 training, 442 testing).

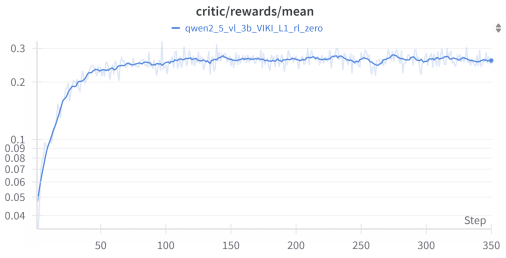
E Analysis of VIKI-R

As illustrated in Figure 5, the four panels reflect two distinct axes of variation: model capacity (3B in panels a–b vs. 7B in c–d) and initialization strategy (RL-only in VIKI-R-ZERO vs. SFT cold-start + RL in VIKI-R). Even without SFT, the 7B R-ZERO curve (panel c) begins at 0.1 and climbs to 0.9 by step 120, compared to 0.04 to 0.27 for the 3B R-ZERO (panel a), underscoring scale effects. However, both R-ZERO variants exhibit sluggish and oscillatory learning: the base model lacks sufficient task reasoning capacity to roll out coherent action sequences, resulting in unstable gradient signals and limited policy improvement without a prior SFT warm-up. Introducing a CoT cold-start further boosts performance: the 3B R variant (panel b) launches at 0.3 and reaches 0.85 by step 140—substantially outpacing its R-ZERO counterpart—while the 7B R (panel d) jumps in at 0.65 and exceeds 0.95 by step 70. Taken together, these results show that both larger capacity and SFT initialization independently accelerate learning, and when combined, yield the fastest convergence and highest final rewards.

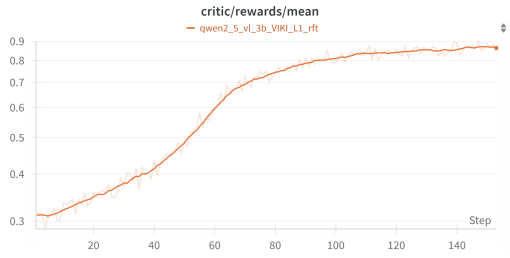
Figure 6 juxtaposes the planning task dynamics across both model sizes and initialization strategies. On the 3B backbone, the RL-only R-ZERO variant (panel a) starts near a negligible mean reward (0.009), briefly peaks at 0.019 around step 50, then settles to 0.011 with persistent fluctuations—indicative of learning but limited headroom. This poor performance stems from the base model's limited reasoning capacity, which fails to produce coherent rollout trajectories without SFT initialization, yielding weak reward signals. In contrast, the CoT-initialized R variant (panel b) launches at 0.56 (reflecting SFT warm-up) and swiftly climbs to 0.92 by step 60, eventually plateauing around 0.94 with minimal oscillation, demonstrating dramatically accelerated and stable policy improvement. For the 7B backbone, R-ZERO (panel c) begins at 0.06 and converges to 0.075 by step 20, maintaining a narrow band around that value thereafter. The 7B R design (panel d), however, initiates at 0.45 and reaches 0.90 by step 60, then settles around 0.92, mirroring the 3B pattern of a strong SFT head start followed by rapid RL fine-tuning. These results confirm that both increased model capacity and SFT cold-start independently enhance planning performance, with their combination yielding the most pronounced gains.

Table 7: Hyperparameter settings for GRPO training

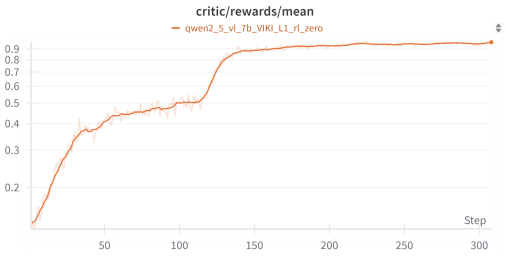
Parameter	**VIKI-L1**	**VIKI-L2**	**VIKI-L3**
Engine	`$1:-vllm`	`$1:-vllm`	`$1:-vllm`
VLLM attention backend	`XFORMERS`	`XFORMERS`	`XFORMERS`
Algorithm (adv estimator)	`grpo`	`grpo`	`grpo`
Train batch size	256	256	256
Max prompt length	4096	4096	4096
Max response length	2048	2048	2048
Filter overlong prompts	True	True	True
Truncation strategy	`error`	`error`	`error`
Actor learning rate	1×10^{-6}	1×10^{-6}	1×10^{-6}
Remove padding	True	True	True
PPO mini-batch size	128	128	128
PPO micro-batch per GPU	10	10	10
Use KL loss	True	True	True
KL loss coefficient	0.01	0.01	0.01
KL loss type	`low_var_kl`	`low_var_kl`	`low_var_kl`
Entropy coefficient	0	0	0
Gradient checkpointing	True	True	True
FSDP param offload	False	False	False
FSDP optimizer offload	False	False	False
Rollout log-prob micro-batch	20	20	20
Tensor model parallel size	1	1	1
Rollout engine name	`vllm`	`vllm`	`vllm`
GPU memory utilization	0.6	0.6	0.6
Chunked prefill	False	False	False
Enforce eager	False	False	False
Free cache engine	False	False	False
Rollout samples (n)	5	5	5
Reference log-prob micro-batch	20	20	20
Reference FSDP param offload	True	True	True
Use KL in reward	False	False	False
Critic warmup steps	0	0	0
GPUs per node	8	8	8
Number of nodes	1	1	1
epoch	5	15	2



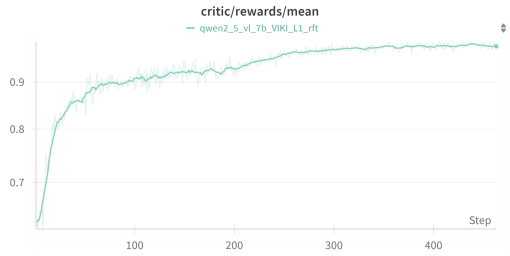
(a) Qwen2.5VL3B-VIKI-R-ZERO



(b) Qwen2.5VL3B-VIKI-R

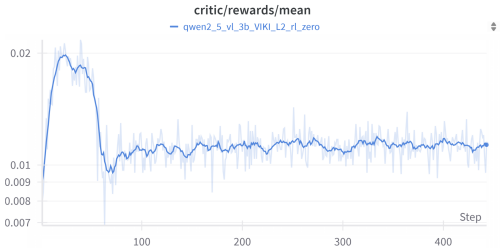


(c) Qwen2.5VL7B-VIKI-R-ZERO

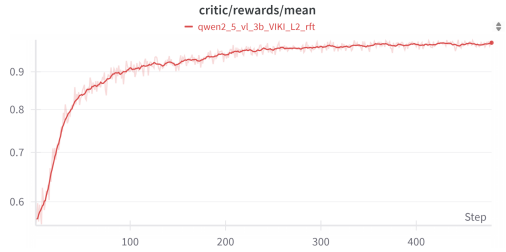


(d) Qwen2.5VL7B-VIKI-R

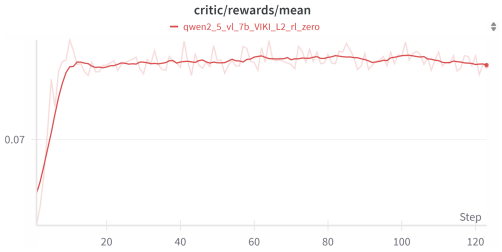
Figure 5: GRPO reward mean curve about task VIKI-L1



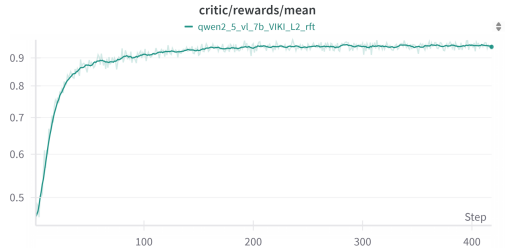
(a) Qwen2.5VL3B-VIKI-R-ZERO



(b) Qwen2.5VL3B-VIKI-R



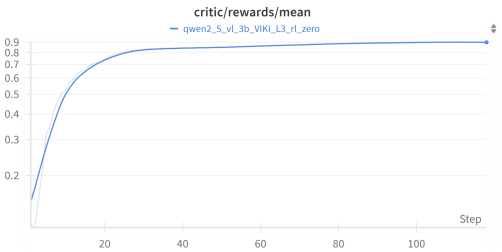
(c) Qwen2.5VL7B-VIKI-R-ZERO



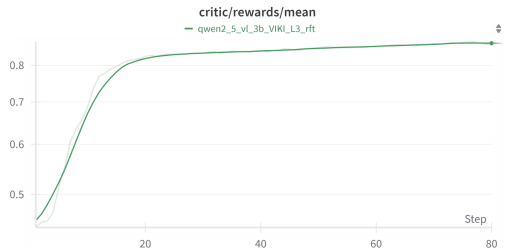
(d) Qwen2.5VL7B-VIKI-R

Figure 6: GRPO reward mean curve about task VIKI-L2

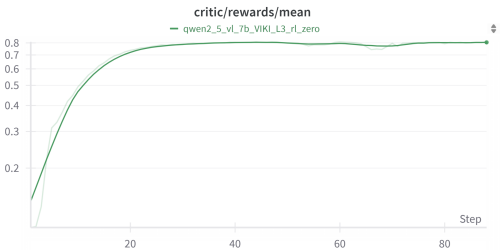
In the trajectory execution task (Fig. 7), both initialization strategies achieve high asymptotic rewards, demonstrating strong local motion capability. On the 3B backbone, R-ZERO rises from ∼0.12 to 0.83, while CoT-initialized R starts higher (∼0.45) and converges faster to 0.84. On the 7B model, both show similar trends, with R-ZERO reaching 0.80 and R stabilizing around 0.75 after a brief dip. Overall, trajectory execution depends less on high-level planning—RL-only training achieves strong performance, while SFT initialization mainly accelerates early convergence.



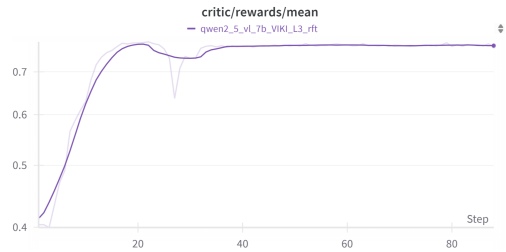
(a) Qwen2.5VL3B-VIKI-R-ZERO



(b) Qwen2.5VL3B-VIKI-R



(c) Qwen2.5VL7B-VIKI-R-ZERO



(d) Qwen2.5VL7B-VIKI-R

Figure 7: GRPO reward mean curve about task VIKI-L3

F Data Demonstration

Fig. 8 provides additional data visualizations showcasing qualitative demos of VIKI-L1 and VIKI-L2.

G Additional Ablation Studies

We further perform four complementary ablation studies to better understand the design choices in **VIKI-R**, focusing on reward structure, model scale, prompt formulation, and policy optimization.

1. Reward Structure. We evaluate the effect of removing the step-level reward in the **VIKI-L2** dataset. As shown in Table 8, removing this component leads to a consistent performance drop on both in-distribution (ID) and out-of-distribution (OOD) splits, confirming that stepwise feedback is crucial for stable reasoning.

Table 8: Effect of step reward on VIKI-L2 performance.

Model	**VIKI-L2-ID (%)**	**VIKI-L2-OOD (%)**
Qwen2.5-VL-7B-Inst (with step)	95.22	33.25
Qwen2.5-VL-7B-Inst (w/o step)	92.06	29.51

2. Model Scaling. We assess the impact of model size using 3B, 7B, and 32B backbones. As Table 9 shows, larger models consistently outperform smaller ones, highlighting that scaling strengthens compositional reasoning and improves both low- and high-level task accuracy.

Table 9: Effect of model scale on reasoning performance.

Model	**VIKI-L1 (%)**	**VIKI-L2 (%)**
Qwen2.5-VL-3B-Inst	74.10	68.78
Qwen2.5-VL-7B-Inst	93.00	69.25
Qwen2.5-VL-32B-Inst	**95.30**	**74.74**

3. Prompt Design. We analyze the effect of removing reasoning guidance prompts such as *"You FIRST think about the reasoning process as an internal monologue and then provide the final answer."* As shown in Tab. 10, accuracy remains nearly unchanged, though convergence becomes slower. This suggests that the reinforcement signal alone enables the model to explore reasoning paths even without explicit instruction.

Table 10: Effect of reasoning prompt on training efficiency.

Model	**VIKI-L1 Acc. (%)**
Qwen2.5-VL-3B-Inst	74.10
Qwen2.5-VL-3B-Inst (w/o guidance)	73.47
Qwen2.5-VL-7B-Inst	93.00
Qwen2.5-VL-7B-Inst (w/o guidance)	92.95

4. Optimization Algorithm. We compare our GRPO-based optimization with the open-source **DAPO** [56] framework. Results in Tab. 11 show minimal accuracy differences, demonstrating that our compositional reinforcement learning pipeline is robust to alternative PPO-style optimizers.

Overall, these studies collectively confirm that **VIKI-R** benefits from stepwise reward shaping and scaling, is largely invariant to prompt formulation, and remains stable across different optimization algorithms.

H Additional Experiments

Extension of Evaluation to Other Relevant Methods. To further contextualize our benchmark, we extended evaluation to include several representative approaches for vision-based multi-agent collaboration, even though none of them directly align with our hierarchical coordination setting.

Table 11: Comparison between GRPO and DAPO on VIKI-L2.

Model	VIKI-L2-ID (%)	VIKI-L2-OOD (%)
Qwen2.5-VL-3B-Inst	93.61	32.11
Qwen2.5-VL-3B-Inst (DAPO)	93.22	33.33
Qwen2.5-VL-7B-Inst	95.22	33.25
Qwen2.5-VL-7B-Inst (DAPO)	95.44	33.58

Specifically, we adapted **ReAct** [51] and the **Multi-Agent Debate (MAD)** framework [10], aligning their input–output protocols to match the VIKI-Bench interface.

Table 12: Evaluation of related methods on VIKI-Bench.

Method	VIKI-L1 (%)	VIKI-L2 (%)
GPT-4o	18.40	17.50
ReAct [51]	18.70	19.88
MAD (2 agents) [10]	**22.52**	**20.54**

The results show that while existing approaches can handle basic visual reasoning and short-horizon interactions, they struggle with long-term coordination and compositional task planning—core aspects emphasized in VIKI-Bench. Our proposed **VIKI-R** framework, combining supervised fine-tuning and reinforcement learning, effectively bridges this gap by supporting structured collaboration among multiple embodied agents.

These findings highlight the growing relevance of multi-robot cooperation and demonstrate how VIKI-Bench fills a critical gap in evaluating hierarchical, vision-based coordination across agents. We expect this benchmark to serve as a foundation for future research on scalable multi-agent reasoning and control.

We report the computational resources and configurations used for fine-tuning **VIKI-R** on Qwen2.5-VL backbones. All experiments were conducted on servers with **8 × NVIDIA A800 GPUs (80GB)**. Each task level (VIKI-L1/L2/L3) was trained separately under identical optimization settings, covering both supervised fine-tuning (SFT) and reinforcement fine-tuning (GRPO). Overall, **VIKI-R** can be trained end-to-end within one day on a single 8-GPU node, ensuring reproducibility.

Table 13: Training resources for Qwen2.5-VL models on different task levels.

Setting	Qwen2.5-VL-7B			Qwen2.5-VL-3B		
	L1	L2	L3	L1	L2	L3
GPU Details	8×A800	8×A800	8×A800	8×A800	8×A800	8×A800
Training Time (h)	22	9	5	13	5.5	3
Training Epochs	5	15	2	5	15	2
Batch Size	256	256	256	256	256	256

I Prompt

Output Constraint

```
Output Format Requirements:
<answer>
  [
    {
      "step": 1,
      "actions": {'R1': ['Move', 'banana'], 'R2': ['Move', 'apple']}
    },
    {
      "step": 2,
```

```
      "actions": {'R1': ['Reach', 'banana'], 'R2': ['Reach', 'apple']}
    }
    # ... subsequent steps ...
  ]
</answer>
- step is the time step number (starting from 1, incrementing sequentially).
- Each robot can only have ONE action per time step.
- "actions" is a dictionary that specifies the action for each robot during a
↪  single time step. Each key (e.g., "R1", "R2") represents a robot. Each
↪  value is a list describing the single action that robot will perform in
↪  this step, with the following format: action_type,
↪  target_object_or_location, (optional: extra_argument)
Action primitives and descriptions: {ACTION_DESCRIPTION}
Available robot set: {robots}
Robot characteristics: {available_robots}
Their available operation APIs: {available_actions}
```

Prompt of VIKI-L1

```
Possible robots: {robot_set}
First, identify the robots visible in the image from a list of "possible
↪  robots". Among the visible robots, select the most suitable one or more
↪  to collaborate on the task.
You FIRST think about the reasoning process as an internal monologue and then
↪  provide the final answer.
The reasoning process MUST BE enclosed within <think> </think> tags. The
↪  final answer MUST BE enclosed within <answer>Your final answer must be
↪  provided as a Python list format, for example: [\'fetch\',
↪  \'unitree_h1\']. Include only the robot names that are suitable for the
↪  task.</answer> tags.
```

Prompt of VIKI-L2

```
You are a plan creator. I will provide you with an image of robots in a
↪  scene, available robots and their action primitives, and a task
↪  description. You need to create a plan to complete the task.
You must first analyze the image to fully understand the scene depicted.
↪  Then, analyze the task description. Finally, create a plan to complete
↪  the task.
Your reasoning must strictly adhere to the visual content of the image and
↪  the task description-no assumptions, hypotheses, or guesses are allowed.
1. Create a plan to complete the task, noting:
   - Each robot can only perform ONE action per time step.
   - Multiple robots can work in parallel, but each robot is limited to one
   ↪  action at a time.
2. You need to first provide your reasoning process within <think> and
↪  </think> tags.
3. Your final answer must be within <answer> and </answer> tags, and
↪  **strictly follow the JSON format specified below**.

Output Format Requirements(please comply strictly, do not output any
↪  additional content):
<answer>
  [
    {{
      "step": 1,
      "actions": {{'R1': ['Move', 'banana'], 'R2': ['Move', 'apple']}}
    }},
    {{
      "step": 2,
```

```
      "actions": {{'R1': ['Reach', 'banana'], 'R2': ['Reach', 'apple']}}
    }}
    # ... subsequent steps ...
  ]
</answer>
Where:
- step is the time step number (starting from 1, incrementing sequentially).
- Each robot can only have ONE action per time step.
- "actions" is a dictionary that specifies the action for each robot during a
↪  single time step. Each key (e.g., "R1", "R2") represents a robot. Each
↪  value is a list describing the single action that robot will perform in
↪  this step, with the following format: action_type,
↪  target_object_or_location, (optional: extra_argument)
Action primitives and descriptions: {ACTION_DESCRIPTION}
Available robot set: {robots}
Robot characteristics: {available_robots}
Their available operation APIs: {available_actions}
```

Prompt of VIKI-L3

```
You are an expert in visual understanding and trajectory planning.
**INPUT:**
* An ego-view image showing two robotic arms working together; the arm
↪  closest to the camera represents **you**.
* A string describing the overall task.
* Two strings specifying your subtask ("you") and your partner's subtask.
**YOUR JOB:**
1. Enclose your scene analysis and task division within `<think>...</think>`
↪  tags.
2. Enclose your final output within `<answer>...</answer>` tags as a nested
↪  list of **ten 2D pixel coordinates**:
   * Two groups of five points each:
     * **First group:** your trajectory
     * **Second group:** your partner's trajectory
3. Follow this format **exactly** (no additional text):
   [[ [x1, y1], [x2, y2], [x3, y3], [x4, y4], [x5, y5] ],
    [ [x1', y1'], [x2', y2'], [x3', y3'], [x4', y4'], [x5', y5'] ]]
```

Primitives and description

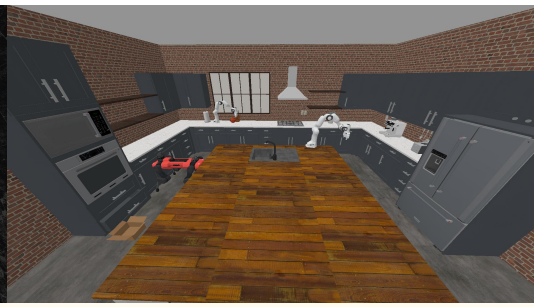
```
ROBOT_DESCRIPTION = {
    'stompy': 'A bipedal robot designed for dynamic walking and stomping
    ↪  tasks, featuring articulated arms. Color: Light blue body with yellow
    ↪  and orange accents.',
    'fetch': 'A wheeled robot with a flexible arm for object manipulation,
    ↪  designed for mobility and dexterity. Color: White with blue and black
    ↪  accents.',
    'unitree_h1': 'A humanoid robot with arms and legs designed for
    ↪  human-like movements and tasks. Color: Black.',
    'panda': 'A fixed robotic arm designed for precise and delicate
    ↪  manipulation tasks. Color: White with black accents.',
    'anymal_c': 'A quadrupedal robot built for navigating rough terrains and
    ↪  performing complex tasks with four articulated legs. Color: Red and
    ↪  black with some accents.',
    'unitree_go2': 'A compact quadrupedal robot optimized for agile movement
    ↪  and stability with four legs for efficient locomotion. Color: White.'
}
ACTION_DESCRIPTION = {
```

```
    'Move': "Command ['Move', 'object']: Robot R moves to the specified
    ↪  object.",
    'Open': "Command ['Open', 'object']: Open the object held by the Robot
    ↪  R's end effector.",
    'Close': "Command ['Close', 'object']: Close the object held by the Robot
    ↪  R's end effector.",
    'Reach': "Command ['Reach', 'object']: Robot R reaches the specified
    ↪  object.",
    'Grasp': "Command ['Grasp', 'object']: Robot R's end effector performs a
    ↪  grasping operation on a specified object.",
    'Place': "Command ['Place', 'object']: Place the object held by the Robot
    ↪  R's end effector at a specified location (the release point, not the
    ↪  object itself).",
    'Push': "Command ['Push', 'object', 'R1']: Robot R pushes the object to
    ↪  robot R1.",
    'Interact': "Command ['Interact', 'object']: A general interaction
    ↪  operation, flexible for representing interactions with any asset."

}
AGENT_AVAIL_ACTIONS = {
    'panda':      ['Reach', 'Grasp', 'Place', 'Open', 'Close', 'Interact'],
    'fetch':      ['Move', 'Reach', 'Grasp', 'Place', 'Open', 'Close',
    ↪  'Interact'],
    'unitree_go2':['Move', 'Push', 'Interact'],
    'unitree_h1': ['Move', 'Reach', 'Grasp', 'Place', 'Open', 'Close',
    ↪  'Interact'],
    'stompy':     ['Move', 'Reach', 'Grasp', 'Place', 'Open', 'Close',
    ↪  'Interact'],
    'anymal_c':   ['Move', 'Push', 'Interact'],
}

AGENT_END_EFFECTOR_NUM = {
    'panda': 1,
    'fetch': 1,
    'unitree_go2': 0,
    'unitree_h1': 2,
    'stompy': 2,
    'anymal_c': 0,
}
```

Task Description: Inspect the table visually. Ensure both the **pear** and the **tomato** are present. Fetch and place the missing fruit, checking all areas including cabinet.

Agent Activation: **R1: fetch**, **R2: stompy**

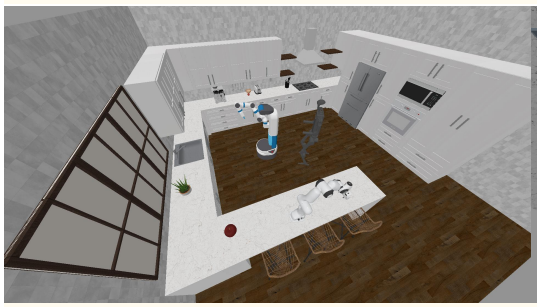
Task Planning:
Step 1: <**R1**, Move tomato>, <**R2**, Move cabinet>
Step 2: <**R1**, Reach tomato>, <**R2**, Reach cabinet>
Step 3: <**R1**, Grasp tomato>, <**R2**, Open cabinet>
Step 4: <**R1**, Move table>, <**R2**, Move pear>
Step 5: <**R1**, Place table>, <**R2**, Reach pear>
Step 6: <**R2**, Grasp pear>
Step 7: <**R2**, Move table>
Step 8: <**R2**, Place table>



Task Description: Get the **peach** over to the sink. In case it's too distant, put the peach into a **cardboardbox**, shift the cardboardbox to the sink, retrieve the peach.

Agent Activation: **R1: panda**, **R2: panda**, **R3: anymal_c**

Task Planning:
Step 1: <**R2**, Reach peach>, <**R3**, Move cardboardbox>
Step 2: <**R2**, Grasp peach>, <**R3**, Push cardboardbox R2>
Step 3: <**R2**, Place cardboardbox>
Step 4: <**R3**, Push cardboardbox R1>
Step 5: <**R1**, Reach peach>
Step 6: <**R1**, Grasp peach>
Step 7: <**R1**, Place sink>



Task Description: Move both the **meat** and the **apple** into the **bowl** to prepare for serving.

Agent Activation: **R1: fetch**, **R2: unitree_h1**

Task Planning:
Step 1: <**R1**, Move apple>, <**R2**, Move meat>
Step 2: <**R1**, Reach apple>, <**R2**, Reach meat>
Step 3: <**R1**, Grasp apple>, <**R2**, Grasp meat>
Step 4: <**R1**, Move bowl>, <**R2**, Move bowl>
Step 5: <**R1**, Place bowl>, <**R2**, Place bowl>

Task Description: Position the **apple** on the **wooden cutting board;** slice it neatly with the **knife**.

Agent Activation: **R1: stompy**, **R2: fetch**

Task Planning:
Step 1: <**R1**, Move tomato>, <**R2**, Move cabinet>
Step 2: <**R1**, Reach tomato>, <**R2**, Reach cabinet>
Step 3: <**R1**, Grasp tomato>, <**R2**, Open cabinet>
Step 4: <**R1**, Move table>, <**R2**, Move pear>
Step 5: <**R1**, Place table>, <**R2**, Reach pear>
Step 6: <**R2**, Grasp pear>
Step 7: <**R2**, Move table>
Step 8: <**R2**, Place table>

Figure 8: Qualitative demonstrations of VIKI-L1 and VIKI-L2 showcasing visual input, agent activation, and task planning across diverse multi-agent collaboration scenarios.