# MECoBench: A Systematic Study of Multi-modal Agent Collaboration in Embodied Environments

**Anonymous authors**
Paper under double-blind review

## Abstract

Recent multimodal large language models (MLLMs) have strong potential as embodied agents, but their ability to collaborate in visually grounded environments remains underexplored. To address this gap, we introduce **MECoBench**, a multimodal embodied cooperation benchmark with a controlled evaluation platform spanning two distinct task domains and diverse task and team settings. Through extensive experiments across various MLLMs, we summarize three key findings: (i) Most models benefit from collaboration, but individual capability alone does not determine collaborative performance. (ii) Communication is essential for collaboration gains, with reliable information sharing and coordinated execution distinguishing effective collaboration from failure. (iii) Appropriate team size, organization, and information sharing can further improve collaborative performance. Generally, MECoBench provides a systematic testbed for understanding the mechanisms and limits of multimodal embodied collaboration.

## 1 Introduction

Recent multimodal large language models (MLLMs) Anthropic (2026); OpenAI (2026); Google DeepMind (2026a) have demonstrated strong vision-language understanding and reasoning capabilities, making them promising foundations for embodied agents in interactive environments Mu et al. (2023); Szot et al. (2025). While most existing MLLM-based studies focus on single-agent embodied intelligence Yang et al. (2025); Liu et al. (2025); Zhang et al. (2025b), many real-world tasks such as household assistance require multiple agents to cooperate Sycara (1998), as shown in Figure 1(a). Previous works on pure language-based multi-agent framework Schmidgall et al.

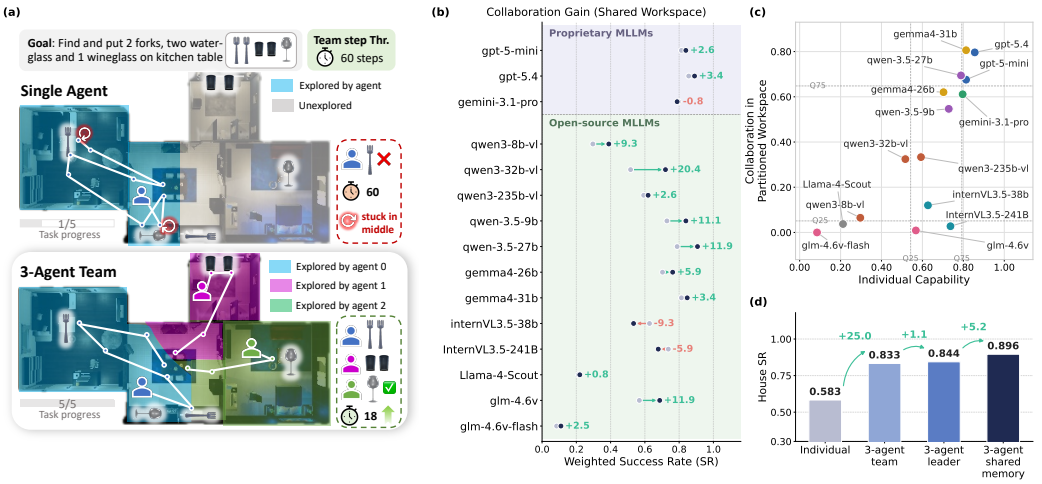



Figure 1: **Multimodal agent collaboration in MECoBench.** (a) A representative household scenario. (b) Changes in success rate with collaboration. (c) Individual capability versus collaborative performance. (d) Better team organization and information sharing further improve collaboration.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

(2025); Hong et al. (2024) have proved that such cooperation could improve efficiency and overcome individual capability limits. However, the potential of multi-agent collaboration under multimodal embodied settings remains an underexplored area.

Answering this is non-trivial, because embodied multi-agent collaboration differs fundamentally from text-based collaboration. Existing multi-agent benchmarks rely on textual task descriptions, shared symbolic states, or pre-processed observations, reducing collaboration to plan coordination or dialogue Zhu et al. (2025); Sun et al. (2025); Zhang et al. (2024a); Yang et al. (2026); Agashe et al. (2025). By contrast, embodied agents must coordinate from partial, dynamically changing visual observations while exploring environments, resolving perceptual inconsistencies, avoiding spatial conflicts, and acting under physical constraints Shridhar et al. (2020). This tightly couples perception, exploration, communication, and coordination, leading to failure modes and emergent behaviors that text-based benchmarks cannot capture Feng et al. (2026). Therefore, it is valuable to systematically study MLLM-based multi-agent collaboration in embodied settings, calling for a new benchmark that supports visually grounded interaction and diverse cooperation patterns.

To address this gap, we introduce MECoBench, a **M**ultimodal **E**mbodied **Co**operation benchmark together with an evaluation platform for systematically analyzing visually grounded embodied multi-agent cooperation. MECoBench spans two distinct task domains: long-horizon household tasks (MECo-House) and outdoor navigation and exploration (MECo-City), covering different spatial scales and interaction requirements. Each task supports Shared and Partitioned Workspace settings, where the latter imposes spatial constraints that create explicit inter-agent dependencies. Beyond task construction, our platform supports controlled variation along multiple dimensions, including team organization, team size and exploration difficulty. This design enables fine-grained analysis of not only whether collaboration helps, but also when, why, and through which mechanisms it improves embodied task solving.

Based on MECoBench, we investigate three questions in sequence: (i) How does performance evolve from individual capability to collaboration? (ii) What drives collaboration gains and failures? (iii) How can team size, organization, and communication mechanism make collaboration more effective? We evaluate a wide range of MLLMs across model families and parameter scales and obtain the following findings:

- **Most models benefit from collaboration, but individual capability does not determine collaborative performance.** Adding a second agent improves task completion for most models, and individual capability is positively correlated with collaborative performance overall (Figure 1 (b), (c)). However, models with similar individual performance can differ substantially when task completion depends on tightly coupled inter-agent coordination.
- **Communication is critical for realizing collaboration gains, but its benefits depend on reliable information sharing and coordinated execution.** Removing communication substantially degrades performance, particularly when coordination is necessary. Effective information sharing and task allocation facilitate successful collaboration, while failures arise from conflicts and execution mismatches among agents and the propagation of unreliable information across the team.
- **Effective team design depends on task demands and model capability.** Moderate team sizes achieve the best balance between collaboration gains and costs, resulting in an inverted-U trend as team size increases. Centralized organization helps many small teams, while shared memory improves performance and efficiency (Figure 1 (d)). Prior-information ablations further show that collaboration can remain useful under missing or noisy locations.

## 2 Related Work

**Multi-Agent System** Inspired by human society, multi-agent systems (MAS) have been studied as a way to improve task efficiency and effectiveness Stone & Veloso (2000).Recent advances in large language models (LLMs) have enabled language-based MAS across diverse domains, including automated research Schmidgall et al. (2025); Su et al. (2025), programming Hong et al. (2024); Islam et al. (2025), medical decision-making Fan et al. (2025); Chen et al. (2025), and social simulation Piao et al. (2025). Other studies further examine their scalability and emergent collaborative behaviors Qian et al.

Table 1: **Comparison with related benchmarks.** Comm.: communication medium. † Only the embodied Minecraft subset of MultiAgentBench is considered.

| Benchmark | Environment | Input | Variable Team Size | Workspace Accessibility | Controlled Difficulty | Comm. |
|---|---|---|---|---|---|---|
| WAH Puig et al. (2021) | Indoor | State | No | No | No | None |
| EmbodiedBench Yang et al. (2025) | Indoor | Multimodal | No | No | No | None |
| COMBO Zhang et al. (2025a) | Indoor | Vision | Yes | No | No | None |
| PARTNR Chang et al. (2025) | Indoor | Text | No | Yes | No | Text |
| MultiAgentBench† Zhu et al. (2025) | Game | Text | No | No | Partial | Text |
| CoELA Zhang et al. (2024a) | Indoor | Text | No | No | No | Text |
| RoCoBench Mandi et al. (2024) | Indoor | Text | No | No | No | Text |
| VillagerBench Dong et al. (2024) | Game | Text | Yes | No | Partial | Text |
| TeamCraft Long et al. (2024) | Game | Multimodal | Yes | No | No | None |
| Collab-Overcooked Sun et al. (2025) | Game | Text | No | No | Yes | Text |
| **MECoBench** | **Indoor/Outdoor** | **Multimodal** | **Yes** | **Yes** | **Yes** | **Text/Vision** |

(2025); Chen et al. (2024). Recent work has also extended these ideas to embodied scenarios Zhang et al. (2025a); Liu et al. (2024).

**Embodied Agent** Recent studies have explored LLM- or MLLM-based embodied agents for a wide range of tasks, including object manipulation Yang et al. (2024); Mon-Williams et al. (2025), navigation Zhang et al. (2024b), complex perception-interaction tasks Szot et al. (2024), and embodied human-AI collaboration Chang et al. (2025). In parallel, several benchmarks have been developed to evaluate embodied agents across diverse environments and capabilities Deitke et al. (2022); Savva et al. (2019); Yang et al. (2025); Cheng et al. (2025). These benchmarks predominantly focus on individual agent, providing limited support for studying collaboration in embodied task solving.

**Embodied Multi-Agent Collaboration Benchmarks** Several benchmarks evaluate collaboration among embodied agents Kang et al. (2025); Puig et al. (2021); Jain et al. (2020); Carroll et al. (2019); Agashe et al. (2025). However, existing benchmarks cover only part of the key challenges in embodied multi-agent collaboration. Some use game environments such as Minecraft or Overcooked Sun et al. (2025); Long et al. (2024), which abstract away many spatial and interaction characteristics of real physical environments. Others mainly focus on indoor household or tabletop scenarios Zhang et al. (2024a; 2025a); Yang et al. (2025), leaving outdoor collaboration at larger spatial scales underexplored. Most existing benchmarks also convert visual observations into text Mandi et al. (2024); Chang et al. (2025); Dong et al. (2024); White et al. (2025), reducing the role of visual grounding and spatial reasoning in collaboration. Existing benchmarks also rarely support controlled studies of team design. MECoBench addresses these gaps with multimodal closed-loop interaction across indoor and outdoor domains and configurable task and team settings. Table 1 summarizes the comparison.

# 3 MECoBench

To study collaboration among MLLM-based agents in embodied environments, we introduce **MECoBench**, a benchmark with a configurable evaluation platform. MECoBench comprises two task domains: **MECo-House** for long-horizon household tasks and **MECo-City** for outdoor navigation and exploration, covering different spatial scales and interaction requirements. We further introduce workspace accessibility constraints to vary coordination requirements. Examples shows in Figure 2.

## 3.1 Benchmark Construction

**Long-Horizon Household Task.** MECo-House evaluates multi-agent collaboration on long-horizon, cross-room object rearrangement in indoor household environments. Each task requires agents to locate target objects, navigate across rooms, and transfer them to designated goal locations. Following WAH Puig et al. (2021), each task goal is defined by a set of predicates describing the desired final state of the environment, e.g., On(`apple`, `kitchentable`). We build MECo-House upon the VirtualHome simulator Puig et al. (2018). To construct realistic task templates, we use an LLM to distill common household activities from existing household-activity corpora, followed by

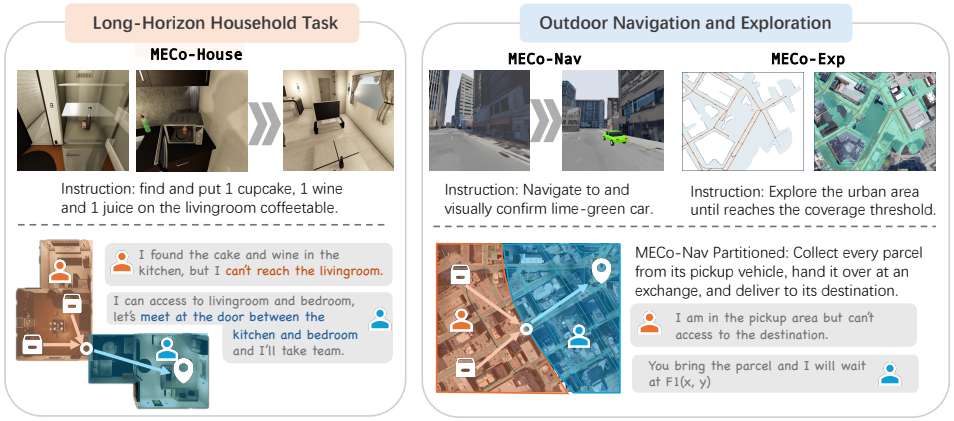



Figure 2: **Task examples in MECoBench.** Partitioned Workspace in MECo-House and MECo-Nav impose accessibility constraints that require agents to coordinate cross-region transfers.

manual review and simulator feasibility checks, resulting in 20 task templates. We further define legal placement rules for task objects to support diverse and realistic distributions. For each task instance, we sample subgoals from a template, ground them to corresponding 3D assets and an feasible apartment scene, and randomly initialize the target objects at legal locations.

**Outdoor Navigation and Exploration.** MECo-City is built on outdoor urban environments and consists of two subsets. **MECo-Nav** focuses on object-goal navigation, where agents search for target cars, navigate to it, and confirm it within 20 meters. **MECo-Exp** focuses on exploration, where agents explore city blocks until map coverage reaches 70%. We build MECo-City on Virtual-Community Zhou et al. (2026) using cropped regions from maps of Detroit and New York, and construct navigation graphs from road and building geometry. We construct diverse instances by varying the region, map layout, and target configuration, while ensuring connectivity and task feasibility.

**Workspace Accessibility.** To model different spatial dependencies in collaboration, we introduce workspace accessibility constraints. In the ***Shared Workspace***, all agents can access all task-relevant regions, enabling parallel execution of task subgoals. In the ***Partitioned Workspace***, agents are assigned disjoint accessible regions, requiring cross-region coordination to complete the task sequentially. For MECo-House, we partition the apartment into disjoint room zones according to the initial and goal locations of task objects. For MECo-Nav, we partition the urban area into disjoint spatial zones and distribute target cars across them. Each zone is assigned to a different agent.

**Data Statistics.** Overall, MECo-House includes 240 base tasks across six difficulty levels, with each comprising 2–7 subgoals. These tasks involve 55 task-relevant objects from 8 categories and span 30 unique apartments. MECo-City includes 100 base tasks, including 60 for MECo-Nav with three difficulty levels and 40 for MECo-Exp. Each MECo-House and MECo-Nav task is instantiated under both Shared and Partitioned Workspace, resulting in 640 tasks in total. Further construction details and statistics are provided in Appendix A.

### 3.2 Evaluation Platform

**Problem Formulation.** We formulate embodied multi-agent cooperation as a finite-horizon decentralized partially observable Markov decision process (Dec-POMDP) Bernstein et al. (2002); Goldman & Zilberstein (2003), specified by $\mathcal{M} = \langle n, \mathcal{S}, G, \{\mathcal{A}_i\}, \{\mathcal{O}_i\}, T, h \rangle$. Here, $n$ denotes the number of agents, $\mathcal{S}$ is the global state space, $G = \{g_1, \ldots, g_K\}$ is the shared set of subgoals, and $h$ is the maximum interaction horizon. Each agent $i$ has an action space $\mathcal{A}_i$ and an observation space $\mathcal{O}_i = \mathcal{O}_i^w \times \mathcal{O}_i^c$, where $\mathcal{O}_i^w$ denotes its local visual observation of the environment and $\mathcal{O}_i^c$ denotes messages received from other agents. At each time step $t$, the agents execute a joint action $\mathbf{a}_t = (a_t^1, \ldots, a_t^n) \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$, and the environment transitions according to $T(s_t, \mathbf{a}_t, s_{t+1}) = P(s_{t+1} \mid s_t, \mathbf{a}_t)$.

**Evaluation Pipeline.** As illustrated in Figure 10 in the appendix, evaluation follows an iterative *observe–communicate–act* workflow. Agents start from their initial positions with a shared task goal.

At each step, each agent integrates its egocentric observation with its action history, recent dialogue, and self-maintained memory, and communicates with teammates according to the specified team organization. By default, we adopt a broadcast protocol, where each agent sends one message to all teammates. Conditioned on its local context and the exchanged messages, each agent selects its next action from the action space described in Appendix B.3. The simulator then executes the joint actions and returns feedback, which is incorporated into their histories and memories for the next step.

**Evaluation Metrics.** We evaluate collaboration from three aspects: **(1) Effectiveness**: *Success Rate (SR)*, defined as the proportion of successful cases, and *Completion Rate (CR)*, defined as the average fraction of completed subgoals across cases. **(2) Efficiency**: *Step-CR AUC (AUC)*, measuring the area under the completion-rate curve over equivalent steps, and *Average Token Cost per Step*. **(3) Collaboration quality**: *Division of Labor (DOL)*, *Conflict Action Rate (CAR)*, and *Handover Failure Rate (HFR)*. DOL measures the distribution of subgoal completion across agents, calculated as $s_i = \text{count}_i / \sum_j \text{count}_j$ and $\text{DOL} = (1 - \sum_i s_i^2)/(1 - 1/n)$, where $n > 1$ is the number of agents, and $s_i$ denotes the fraction of subgoals completed by agent $i$ among all completed subgoals. CAR measures the average fraction of conflicting actions between agents, such as grabbing or opening the same object. HFR measures the average fraction of failed handovers in sequential tasks.

# 4 Experiments

## 4.1 Experimental Setup

Due to the computational cost of multi-agent evaluation, we use a fixed subset of 226 tasks: 96 MECo-House and 12 MECo-Nav tasks under both workspace settings, and 10 MECo-Exp tasks in Shared Workspace. Shared Workspace evaluations use the same total action budgets $B$ for single agent and multi-agent teams, i.e., $B/n$ steps for each agent in an $n$-agent team. Partitioned Workspace is evaluated with a fixed two-agent teams. Detailed task-specific budgets are provided in Appendix B.1.

**Evaluated Models** We evaluate a diverse set of both proprietary and open-source models. The proprietary models include GPT-5.4 OpenAI (2026), Gemini-3.1-Pro Google DeepMind (2026a), and GPT-5-mini OpenAI (2025) The open-source models include Qwen3-VL Bai et al. (2025), Qwen3.5 Qwen Team (2026), Gemma4 Google DeepMind (2026b), InternVL 3.5 Wang et al. (2025), Llama4 Meta AI (2025) and GLM 4.6V Team et al. (2025), covering model sizes from 8B to 235B parameters. Appendix B.2 lists all models and generation configurations used in the evaluation.

## 4.2 Main Results: From Individual Capability to Collaborative Performance

We first evaluate each model's individual capability, and then introduce a second agent into the same shared workspace to examine whether collaboration helps task completion. We further impose spatial access constraints and evaluate two-agent teams in the Partitioned Workspace, where successful task completion requires tighter inter-agent coordination. Results are shown in Table 2.

**Most models benefit from collaboration across both household and outdoor tasks.** Introducing a second agent to the shared workspace improves performance for the majority of models across all tasks. Figure 1 (b) reports the case-weighted change in SR across the full evaluation set. The improvement is particularly large for weak and mid-level models, likely because their single-agent baselines leave more room for gains. However, these benefits are not universal. The InternVL3.5 series degrades on MECo-House, with InternVL3.5-38B also dropping on MECo-Nav. Other strong models, such as Gemma4-31B and GPT-5.4, show little or no improvement on some tasks. Overall, collaboration is generally beneficial, but its gains vary substantially across models.

To understand this variation, we further examine the relationship between individual and collaborative capability. We compare individual capability with collaborative performance on partitioned-workspace tasks. As shown in Figure 1 (c), **individual capability is positively correlated with collaborative performance, but does not fully determine it.** Stronger models generally remain stronger in collaboration, yet clear exceptions exist. InternVL3.5 performs competitively individually but struggles in collaboration, while Qwen3-VL shows substantial gains in individual capability from 32B to 235B but only limited improvement in collaborative performance. These results reveal a gap between individual execution and effective group collaboration.

Table 2: **Individual capability and collaborative performance across models.** ↑ and ↓ indicate increases and decreases relative to the corresponding individual capability; ∼ indicates no change. Bold indicates the best result in each column.

| Model | Individual Capability | | | | Shared Workspace | | | | Partitioned Workspace | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | House | | Nav | Exp | House | | Nav | Exp | House | | Nav | |
| | SR | CR | SR | SR | SR | CR | SR | SR | SR | CR | SR | CR |
| *Proprietary MLLMs* | | | | | | | | | | | | |
| GPT-5-mini | **0.906** | **0.974** | 0.333 | 0.500 | 0.917$^{\uparrow}$ | 0.982$^{\uparrow}$ | 0.417$^{\uparrow}$ | 0.600$^{\uparrow}$ | 0.719 | 0.901 | 0.330 | 0.597 |
| GPT-5.4 | 0.854 | 0.960 | **1.000** | **0.700** | 0.896$^{\uparrow}$ | 0.969$^{\uparrow}$ | **1.000**$^{\sim}$ | 0.700$^{\sim}$ | 0.771 | **0.914** | **1.000** | **1.000** |
| Gemini-3.1-Pro | 0.802 | 0.927 | **1.000** | 0.500 | 0.771$^{\downarrow}$ | 0.937$^{\uparrow}$ | **1.000**$^{\sim}$ | 0.700$^{\uparrow}$ | 0.573 | 0.824 | 0.917 | 0.931 |
| *Open-source MLLMs* | | | | | | | | | | | | |
| Qwen3-8B-VL | 0.354 | 0.620 | 0.000 | 0.100 | 0.458$^{\uparrow}$ | 0.770$^{\uparrow}$ | 0.083$^{\uparrow}$ | 0.100$^{\sim}$ | 0.073 | 0.332 | 0.000 | 0.264 |
| Qwen3-32B-VL | 0.583 | 0.753 | 0.083 | 0.400 | 0.802$^{\uparrow}$ | 0.934$^{\uparrow}$ | 0.167$^{\uparrow}$ | 0.600$^{\uparrow}$ | 0.365 | 0.620 | 0.000 | 0.319 |
| Qwen3-235B-VL | 0.656 | 0.855 | 0.417 | 0.200 | 0.667$^{\uparrow}$ | 0.896$^{\uparrow}$ | 0.250$^{\downarrow}$ | 0.600$^{\uparrow}$ | 0.354 | 0.675 | 0.167 | 0.508 |
| Qwen3.5-9B | 0.854 | 0.924 | 0.250 | 0.100 | **0.938**$^{\uparrow}$ | **0.985**$^{\uparrow}$ | 0.583$^{\uparrow}$ | 0.200$^{\uparrow}$ | 0.615 | 0.844 | 0.000 | 0.375 |
| Qwen3.5-27B | 0.844 | 0.950 | 0.667 | 0.400 | **0.938**$^{\uparrow}$ | 0.969$^{\uparrow}$ | 0.917$^{\uparrow}$ | 0.600$^{\uparrow}$ | 0.698 | 0.872 | 0.667 | 0.806 |
| Gemma4-26B | 0.760 | 0.867 | 0.500 | 0.400 | 0.812$^{\uparrow}$ | 0.944$^{\uparrow}$ | 0.500$^{\sim}$ | 0.600$^{\uparrow}$ | 0.688 | 0.839 | 0.080 | 0.417 |
| Gemma4-31B | 0.833 | 0.921 | **1.000** | 0.400 | 0.833$^{\sim}$ | 0.945$^{\uparrow}$ | **1.000**$^{\sim}$ | **0.800**$^{\uparrow}$ | **0.823** | 0.904 | 0.670 | 0.842 |
| InternVL-3.5-38B | 0.677 | 0.875 | 0.250 | 0.600 | 0.594$^{\downarrow}$ | 0.830$^{\downarrow}$ | 0.000$^{\downarrow}$ | 0.600$^{\sim}$ | 0.135 | 0.429 | 0.000 | 0.139 |
| InternVL-3.5-241B | 0.833 | 0.946 | 0.000 | **0.700** | 0.719$^{\downarrow}$ | 0.914$^{\downarrow}$ | 0.333$^{\uparrow}$ | 0.700$^{\sim}$ | 0.031 | 0.267 | 0.000 | 0.292 |
| Llama-4 | 0.250 | 0.467 | 0.000 | 0.100 | 0.250$^{\sim}$ | 0.543$^{\uparrow}$ | 0.000$^{\sim}$ | 0.200$^{\uparrow}$ | 0.042 | 0.225 | 0.000 | 0.194 |
| GLM-4.6V | 0.594 | 0.825 | 0.333 | 0.600 | 0.719$^{\uparrow}$ | 0.895$^{\uparrow}$ | 0.333$^{\sim}$ | **0.800**$^{\uparrow}$ | 0.010 | 0.283 | 0.000 | 0.361 |
| GLM-4.6V-Flash | 0.094 | 0.351 | 0.000 | 0.100 | 0.125$^{\uparrow}$ | 0.456$^{\uparrow}$ | 0.000$^{\sim}$ | 0.100$^{\sim}$ | 0.000 | 0.090 | 0.000 | 0.361 |

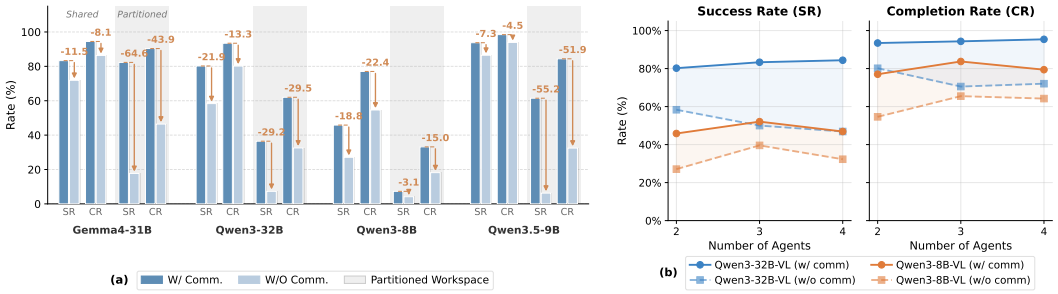



Figure 3: **Ablation study on communication.** (a) Performance changes when communication is removed. (b) Shared-workspace performance with and without communication across team sizes.

### 4.3 What Drives Collaboration Gains and Failures?

Individual capability alone does not fully explain collaborative performance, suggesting that collaboration depends critically on inter-agent interaction rather than simply on the aggregation of individual behaviors. We first examine the role of communication, and then identify effective collaborative behaviors and analyze failure cases for their underlying causes.

#### 4.3.1 Role of Communication

We ablate communication from the multi-agent team while retaining the same team size and task-specific action budget. Figure 3 (a) shows that removing communication consistently degrades performance across models. For Qwen3-32B-VL, two non-communicating agents achieve 58.3% SR in a shared workspace, matching the single-agent baseline; enabling communication raises SR to 80.2%. Thus, **additional agents alone do not account for the observed collaboration gains**. The drop is much larger on partitioned workspace. When the number of agents grows, the gap between communication and no communication widens. These results indicate that **communication is essential for effective collaboration**, particularly under stronger coordination demands and larger teams. More details are provided in Appendix D.2.

Table 3: **Collaborative behaviors across all models.** Freq.: cases containing each behavior (%). ΔSR/CR: with minus without (pp). WS: workspace.

| Behav. | Shared WS | | | Partitioned WS | | |
|---|---|---|---|---|---|---|
| | Freq. | ΔSR | ΔCR | Freq. | ΔSR | ΔCR |
| Info-Plan | 75.6 | +4.6 | +2.8 | 86.3 | +16.9 | +18.0 |
| Act-Del. | 57.1 | +12.2 | +7.7 | 54.3 | +13.1 | +19.2 |
| Info-Loc. | 47.3 | +9.1 | +6.9 | 68.0 | +7.6 | +13.3 |
| Info-Obj. | 25.9 | +5.0 | +3.9 | 16.3 | −6.7 | −1.6 |
| Task-Assign | 15.7 | +1.9 | +2.9 | 8.5 | −11.5 | −4.9 |

Table 4: **Failure mechanisms.** F/S: frequency in failed/successful episodes (%). ΔSR: with minus without (pp).

| Behav. | Shared WS | | Partitioned WS | |
|---|---|---|---|---|
| | F/S | ΔSR | F/S | ΔSR |
| Act-Conf. | 20.2/15.9 | −6.1 | 0.0/0.0 | — |
| Cmt-Mis. | 17.7/7.4 | −20.9 | 11.0/6.3 | −16.6 |
| Dep-Fail. | 0.0/0.0 | — | 46.6/47.8 | −2.5 |
| Rep-Err. | 18.1/6.5 | −18.1 | 31.9/18.4 | −21.3 |
| Msg-Prop. | 4.4/1.2 | −23.2 | 10.9/5.7 | −20.9 |

#### 4.3.2 What Behaviors Support Effective Collaboration?

We delve deeper into fine-grained behaviors in communication messages and action trajectories using keyword-based matching, reported in Table 3. Examples and further details are in Appendix E.

**Information sharing.** Agents share their intended plans (*Info-Plan*), current locations (*Info-Loc*), and detected task-relevant objects (*Info-Obj*). Plan and location sharing are frequent and consistently beneficial, showing that **it is important to keep teammates informed about intended actions and current state**. Object reports alone shows mixed effects, suggesting that identifying a target does not ensure that teammates agree on who will act on it or whether the subgoal is complete.

**Task Coordination.** Action delegation (*Act-Del.*) converts shared information into specific requests to grab, check, or place an object, and brings the strongest gains in both shared and partitioned workspace. In contrast, task assignment (*Task-Assign*) provides coarser responsibility divisions, such as assigning rooms, areas, or subtasks, and offers only marginal benefit in shared workspace while hurting partitioned ones. **More specific and actionable communication is easier to translate into successful execution**, while a division of labor is insufficient if it does not respect spatial access and execution order.

#### 4.3.3 Where Does Collaboration Break Down?

We identify two major failure modes failure modes in multi-agent collaboration and further analyze five concrete mechanisms underlying these failures. Table 4 summarizes their frequencies and associations with task success, while Appendix F provides examples and additional analysis.

**Multiple agents can introduce conflicts and execution mismatches.** Poor coordination manifests as overlapping work, unfulfilled commitments, and infeasible handovers. Failed shared workspace episodes exhibit more conflict actions (*Act-Conf.*). Commitment–execution mismatch (*Cmt-Mis.*), where agents fail to follow their declared plans, is associated with markedly lower SR in both settings. Partitioned workspace tasks additionally expose dependency coordination failures (*Dep-Fail.*) when agents attempt handovers without confirming location, object possession, or a free hand. Together, these results show that effective collaboration requires consistency across task allocation, communicated commitments, and actual execution.

**Unreliable information can propagate across agents.** State-inconsistent reporting (*Rep-Err.*) includes incorrect claims about location, possession, placement, or task completion and is more frequent in failed episodes in both settings. Message-linked error propagation (*Msg-Prop.*) extends an individual error into team state when teammates incorporate wrong message content into memory. Although less frequent, it has a substantial negative association with task completion. These findings highlight that reliable collaboration requires accurate state reports and belief verification mechanisms.

## 5 Improving Collaborative Performance

Motivated by the preceding analysis, we explore how to further improve collaboration by enhancing task coordination and information sharing. Specifically, we study whether collaboration can be strengthened by **scaling the team size**, **structuring team organization** for better task allocation and

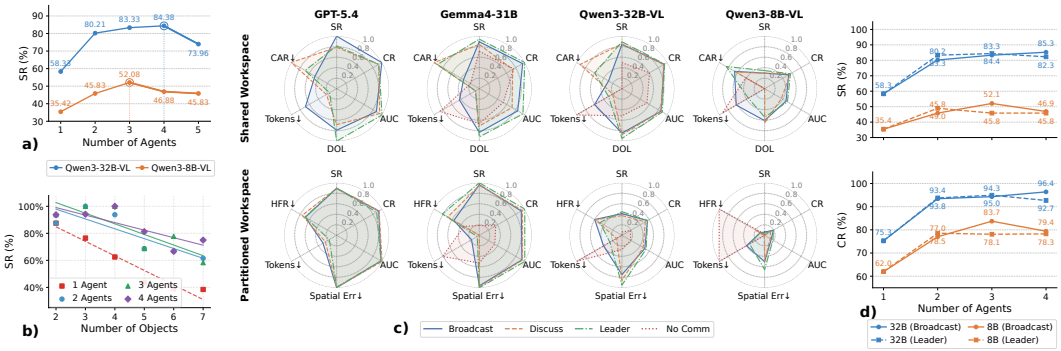



Figure 4: **Effects of team size, task complexity, and team organization.** (a) SR across team sizes on shared workspace. (b) SR of Qwen3-32B-VL versus the number of target objects, with fitted trends for different team sizes. (c) Performance profiles of two-agent teams under different team organizations. (d) SR and CR across team sizes under decentralized and centralized teams.

coordination, and **improving communication mechanisms** to preserve and propagate information more effectively.

### 5.1 Do More Agents Help?

We vary the number of agents from one to five on parallel tasks while keeping the total action budget fixed. Figure 4(a) shows an inverted-U-shaped trend: adding agents initially improves performance, but further scaling leads to saturation and then degradation. This indicates that **team size must balance additional collaboration opportunities against coordination costs**.

The balance also depends on task complexity. Using object count as a proxy, Figure 4(b) shows that success rates decline with increasing object count, due to longer trajectories and greater planning difficulty (see Appendix D.6 for full results across all models and task conditions). Nevertheless, **multi-agent settings maintain more robust performance under high-complexity tasks**, highlighting the value of collaboration in complex embodied scenarios.

### 5.2 Exploring the Effect of Team Organization

**Team Organization.** We further investigate whether a better team organization can improve task allocation and execution coordination. In addition to the default *broadcast* setting, we introduce *discussion* and *leader–worker*, and group the three organizations into decentralized and centralized structures. **Decentralized Team** enables peer-level coordination through *broadcast* or *discussion*. In discussion, agents communicate sequentially and can continue for another round when consensus is not reached, providing more opportunities to coordinate task allocation and avoid conflicts. **Centralized Team** assigns one agent as *leader* to aggregate other *workers* reports and assign actions to the team. This centralized structure provides a global view of team progress and observations, helping align agents' efforts and reduce redundant or conflicting actions.

Figure 4(c) shows that leader-worker achieves the best overall performance for most models, suggesting that **centralized collaboration helps small teams allocate tasks and avoid spatial conflicts**. Its improvements in division of labor and reductions in spatial errors are consistent with clearer task allocation. Discussion reduces action conflicts but does not consistently improve overall success. However, GPT-5.4 remains competitive with simple broadcast communication, indicating that stronger models may rely less on structured collaboration mode.

Figure 4(d) further shows that centralized team saturates earlier as teams grow, while decentralized team becomes more competitive by avoiding the bottleneck of relying on a single leader to aggregate information and assign actions. Leader model capability also matters, as Qwen3-32B-VL benefits from three agents, whereas Qwen3-8B-VL peaks at two agents and then degrades. Therefore, **the best collaboration mode depends on both team size and model capability**. We further conduct experiments showing that a strong leader indeed improves task completion, while the workers' ability to correctly execute is also critical (Results in Appendix D.4).

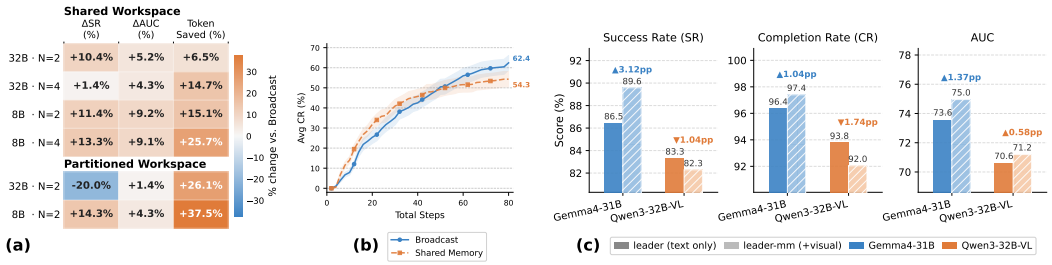



Figure 5: **Shared memory and visual augmentation in collaboration.** (a) Relative changes in effectiveness and efficiency of Qwen3-VL with shared memory over broadcast. (b) Average completion progress over steps under broadcast and shared memory for Qwen3-32B-VL. (c) Leader-based collaboration with and without visual augmentation.

### 5.3 Exploring Alternative Communication Mechanisms

Text messages can leave task state fragmented across exchanges and may omit visually grounded details. We therefore explore two alternative communication mechanisms: *shared memory* and *vision-augmented leadership*.

**Shared memory leads to efficient and effective collaboration.** Agents replace explicit message exchange with a shared memory block. As shown in Figure 5(a), this mechanism improves the performance and greatly reduces token consumption, indicating more efficient collaboration (absolute results in Figure 21). It consistently improves success rate on shared workspace tasks, but degrades on some partitioned workspace tasks, where precise timing and spatial alignment may still require explicit communication. Nevertheless, shared memory accelerates early progress (Figure 5(b)), reducing more conflicts than broadcast under smaller step budgets.

**Visual communication helps when the leader is capable.** To give the leader more grounded context, in vision-augmented leadership, workers send their current visual observations alongside textual reports. As shown in Figure 5(c), additional visual information improves AUC for both models, while its impact on SR and CR varies across models. This difference may stem from models' different abilities to effectively use visual cues. Models with stronger visual capabilities (Gemma4-31B), benefit from the additional visual input, whereas weaker models (Qwen3-32B-VL) drop.

The team-size and communication results suggest that well-organized collaboration can help a smaller model make fuller use of its capabilities. Combining these factors, we progressively adopt the most effective team configurations for Qwen3-VL-32B. As shown in Figure 1 (d), its SR increases from 58.3% to 89.6%, surpassing the single-agent performance of most strong models including GPT-5.4.

To complement the results in Sections 4 and 5, we further examine sensitivity to prior information, dialogue window size, and prompt design, as well as cross-simulator generalization in Appendix G.

## 6 Conclusion

We introduced **MECoBench**, a novel multimodal embodied cooperation benchmark spanning long-horizon household tasks and outdoor navigation and exploration with a configurable evaluation platform. Across a broad range of models, we find that collaboration benefits most models, yet individual capability alone does not determine collaborative performance. Communication is critical, while reliable information sharing and coordinated execution distinguish successful teams from failures caused by conflicts, execution mismatches, and error propagation. Effective team design can further improve collaborative performance. These findings show that advancing embodied multi-agent systems requires not only stronger individual models, but also more reliable mechanisms for communication and coordination. Overall, our study provides deeper insights into MLLM-based embodied collaboration and establishes MECoBench as a systematic testbed for evaluating the collaborative capabilities of future MLLMs.

## LIMITATIONS

In this work, we propose MECoBench and conduct systematic experiments to study cooperative behaviors of MLLM-based agents in embodied environments. Our analysis reveals several non-trivial patterns in multi-agent collaboration, but there remain several limitations for future work.

**Environment and embodiment coverage.** Although MECoBench extends beyond household settings to outdoor urban environments, broader embodied environments and interaction settings remain to be explored. Our evaluation remains simulation-based and does not include real-robot experiments. Moreover, given the limited low-level control capabilities of current MLLMs, MECoBench primarily focuses on high-level decision making and coordination rather than low-level action control. Future work can incorporate richer interaction tasks and real-world robotic settings to further assess the generality of the observed collaboration patterns.

**Evaluation scale.** MECoBench release a large scale and diverse dataset, but exhaustively evaluating every model and collaboration configuration remains computationally expensive due to the cost of long-horizon multi-agent interaction. We therefore conduct experiments on representative task subsets. Future work can extend evaluation to broader task coverage and larger model sets.

**Agent scaling.** Due to spatial constraints in the environment and the complexity of task execution, our experiments scale the team size up to 5 agents. Larger teams and more complex tasks are therefore left unexplored.

## AI USE STATEMENT

Our benchmark evaluation include GPT-5.4 OpenAI (2026), Gemini-3.1-Pro Google DeepMind (2026a), GPT-5-mini OpenAI (2025), Qwen3-VL-8B, 32B and 235B Bai et al. (2025), Qwen3.5-9B and 27B Qwen Team (2026), Gemma4-31B-it and 26B Google DeepMind (2026b), InternVL 3.5-38B and 241B Wang et al. (2025), Llama4-Scout Meta AI (2025) and GLM 4.6V and 4.6V-Flash Team et al. (2025).

Generative AI tools are used to assist with language editing, literature search and summarization. All AI-assisted outputs were critically reviewed and revised by the authors.

## ETHICS STATEMENT

This work studies multi-agent collaboration entirely in simulated environments and does not involve human subjects or personally identifiable information. The benchmark is intended for research purposes, and its results should not be directly interpreted as evidence of reliable or safe performance in real-world deployments.

## REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the benchmark construction and task generation process in Appendix A, and the evaluation protocol, action space, and metrics in Section 3.2 and Appendix B.3. Implementation details, including simulator configurations, model settings, prompts, and evaluation procedures, are documented in Appendices B.1, B.2, and C. Additional experimental results and analyses are provided in Appendices D.1, E, F, and G. Upon acceptance, we will publicly release the benchmark data and code to facilitate reproduction and future research.

## REFERENCES

Saaket Agashe, Yue Fan, Anthony Reyna, and Xin Eric Wang. LLM-coordination: Evaluating and analyzing multi-agent coordination abilities in large language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 8053–8072, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.448. URL https://aclanthology.org/2025.findings-naacl.448/.

Anthropic. Claude Opus 4.7 System Card. https://www.anthropic.com/system-cards, April 2026. System card PDF available at https://cdn.sanity.io/files/4zrzovbb/website/037f06850df7fbe871e206dad004c3db5fd50340.pdf. Accessed May 23, 2026.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research*, 27(4):819–840, November 2002. ISSN 1526-5471. doi: 10.1287/moor.27.4.819.297. URL http://dx.doi.org/10.1287/moor.27.4.819.297.

Micah Carroll, Rohin Shah, Mark Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-ai coordination. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/f5b1b89d98b7286673128a5fb112cb9a-Paper.pdf.

Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Mottaghi, Priyam Parashar, Siddharth Patki, Ishita Prasad, Xavier Puig, Akshara Rai, Ram Ramrakhya, Daniel Tran, Joanne Truong, John Turner, Eric Undersander, and Tsung-Yen Yang. Partnr: A benchmark for planning and reasoning in embodied multi-agent tasks. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 65205–65268, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a3cf318fbeec1126da21e9185ae9908c-Paper-Conference.pdf.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 20094–20136, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/578e65cdee35d00c708d4c64bce32971-Paper-Conference.pdf.

Xi Chen, Huahui Yi, Mingke You, WeiZhi Liu, Li Wang, Hairui Li, Xue Zhang, Yingman Guo, Lei Fan, Gang Chen, Qicheng Lao, Weili Fu, Kang Li, and Jian Li. Enhancing diagnostic capability with multi-agents conversational large language models. *npj Digital Medicine*, 8(1):159, 2025. ISSN 2398-6352. doi: 10.1038/s41746-025-01550-0. URL https://doi.org/10.1038/s41746-025-01550-0.

Zhili Cheng, Yuge Tu, Ran Li, Shiqi Dai, Jinyi Hu, Shengding Hu, Jiahao Li, Yang Shi, Tianyu Yu, Weize Chen, Lei Shi, and Maosong Sun. Embodiedeval: Evaluate multimodal llms as embodied agents, 2025. URL https://arxiv.org/abs/2501.11858.

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied ai using procedural generation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 5982–5994. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/27c546ab1e4f1d7d638e6a8dfbad9a07-Paper-Conference.pdf.

Yubo Dong, Xukun Zhu, Zhengzhe Pan, Linchao Zhu, and Yi Yang. VillagerAgent: A graph-based multi-agent framework for coordinating complex task dependencies in Minecraft. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 16290–16314, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.964. URL https://aclanthology.org/2024.findings-acl.964/.

Zhihao Fan, Lai Wei, Jialong Tang, Wei Chen, Wang Siyuan, Zhongyu Wei, and Fei Huang. AI hospital: Benchmarking large language models in a multi-agent medical interaction simulator. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 10183–10213, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.680/.

Zhaohan Feng, Ruiqi Xue, Lei Yuan, Yang Yu, Ning Ding, Meiqin Liu, Bingzhao Gao, Jian Sun, Xinhu Zheng, and Gang Wang. Multi-agent embodied ai: advances and future directions. *Science China Information Sciences*, 69(5):151202, 2026. ISSN 1869-1919. doi: 10.1007/s11432-025-4820-4. URL https://doi.org/10.1007/s11432-025-4820-4.

Claudia V. Goldman and Shlomo Zilberstein. Optimizing information exchange in cooperative multi-agent systems. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, pp. 137–144, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 1581136838. doi: 10.1145/860575.860598. URL https://doi.org/10.1145/860575.860598.

Google DeepMind. Gemini 3.1 Pro Model Card. https://deepmind.google/models/model-cards/gemini-3-1-pro/, February 2026a. Published February 2026. Accessed May 23, 2026.

Google DeepMind. Gemma 4 model card. https://ai.google.dev/gemma/docs/core/model_card_4, 2026b. Accessed: 2026-05-26.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, zili wang, Steven Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. Metagpt: Meta programming for a multi-agent collaborative framework. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 23247–23275, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/6507b115562bb0a305f1958ccc87355a-Paper-Conference.pdf.

Md. Ashraful Islam, Mohammed Eunus Ali, and Md Rizwan Parvez. CodeSim: Multi-agent code generation and problem solving through simulation-driven planning and debugging. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5128–5154, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.285. URL https://aclanthology.org/2025.findings-naacl.285/.

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 Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, pp. 471–490, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58558-7.

Li Kang, Xiufeng Song, Heng Zhou, Yiran Qin, Jie Yang, Xiaohong Liu, Philip Torr, LEI BAI, and Zhenfei Yin. Viki-r: Coordinating embodied multi-agent cooperation via reinforcement learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/ab16cc8f472cf94b282dbff7b99f5f8e-Paper-Datasets_and_Benchmarks_Track.pdf.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model

serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Xiao Liu, Tianjie Zhang, Yu Gu, Iat Long Iong, Song XiXuan, Yifan Xu, Shudan Zhang, Hanyu Lai, Jiadai Sun, Xinyue Yang, Yu Yang, Zehan Qi, Shuntian Yao, Xueqiao Sun, Siyi Cheng, Qinkai Zheng, Hao Yu, Hanchen Zhang, Wenyi Hong, Ming Ding, Lihang Pan, Xiaotao Gu, Aohan Zeng, Zhengxiao Du, Chan Hee Song, Yu Su, Yuxiao Dong, and Jie Tang. Visualagentbench: Towards large multimodal models as visual foundation agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=2snKOc7TVp.

Xinzhu Liu, Di Guo, Xinyu Zhang, and Huaping Liu. Heterogeneous embodied multi-agent collaboration. *IEEE Robotics and Automation Letters*, 9(6):5377–5384, 2024. doi: 10.1109/LRA.2024.3390588.

Qian Long, Zhi Li, Ran Gong, Ying Nian Wu, Demetri Terzopoulos, and Xiaofeng Gao. Teamcraft: A benchmark for multi-modal multi-agent systems in minecraft, 2024. URL https://arxiv.org/abs/2412.05255.

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)*, pp. 286–299, 2024. doi: 10.1109/ICRA57147.2024.10610855.

Meta AI. Llama 4: Model cards and prompt formats. https://www.llama.com/docs/model-cards-and-prompt-formats/llama4/, 2025. Accessed: 2026-05-26.

Ruaridh Mon-Williams, Gen Li, Ran Long, Wenqian Du, and Christopher G. Lucas. Embodied large language models enable robots to complete complex tasks in unpredictable environments. *Nature Machine Intelligence*, 7(4):592–601, 2025. doi: 10.1038/s42256-025-01005-x.

Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. Embodiedgpt: Vision-language pre-training via embodied chain of thought. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 25081–25094. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/4ec43957eda1126ad4887995d05fae3b-Paper-Conference.pdf.

OpenAI. Gpt-5 mini model. https://developers.openai.com/api/docs/models/gpt-5-mini, 2025. Accessed: 2026-05-26.

OpenAI. Gpt-5.4 thinking system card. https://openai.com/index/gpt-5-4-thinking-system-card/, March 2026. Accessed: 2026-05-26.

Jinghua Piao, Yuwei Yan, Jun Zhang, Nian Li, Junbo Yan, Xiaochong Lan, Zhihong Lu, Zhiheng Zheng, Jing Yi Wang, Di Zhou, Chen Gao, Fengli Xu, Fang Zhang, Ke Rong, Jun Su, and Yong Li. Agentsociety: Large-scale simulation of llm-driven generative agents advances understanding of human behaviors and society. Available at SSRN, 2025. URL https://ssrn.com/abstract=5954414.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

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. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=w_7JMpGZRh0.

Chen Qian, Zihao Xie, YiFei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Scaling large language model-based multi-agent collaboration. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 41488–41505, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/66a026c0d17040889b50f0dfa650e5e0-Paper-Conference.pdf.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. Agent laboratory: Using LLM agents as research assistants. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 5977–6043, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.320. URL https://aclanthology.org/2025.findings-emnlp.320/.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Peter Stone and Manuela Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, 2000. doi: 10.1023/A:1008942012299. URL https://doi.org/10.1023/A:1008942012299.

Haoyang Su, Renqi Chen, Shixiang Tang, Zhenfei Yin, Xinzhe Zheng, Jinzhe Li, Biqing Qi, Qi Wu, Hui Li, Wanli Ouyang, Philip Torr, Bowen Zhou, and Nanqing Dong. Many heads are better than one: Improved scientific idea generation by a LLM-based multi-agent system. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28201–28240, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1368. URL https://aclanthology.org/2025.acl-long.1368/.

Haochen Sun, Shuwen Zhang, Lujie Niu, Lei Ren, Hao Xu, Hao Fu, Fangkun Zhao, Caixia Yuan, and Xiaojie Wang. Collab-overcooked: Benchmarking and evaluating large language models as collaborative agents. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4922–4951, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.249. URL https://aclanthology.org/2025.emnlp-main.249/.

Katia P Sycara. Multiagent systems. *AI magazine*, 19(2):79–79, 1998.

Andrew Szot, Max Schwarzer, Harsh Agrawal, Bogdan Mazoure, Rin Metcalf, Walter Talbott, Natalie Mackraz, R Devon Hjelm, and Alexander T Toshev. Large language models as generalizable policies for embodied tasks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=u6imHU4Ebu.

Andrew Szot, Bogdan Mazoure, Omar Attia, Aleksei Timofeev, Harsh Agrawal, Devon Hjelm, Zhe Gan, Zsolt Kira, and Alexander Toshev. From multimodal llms to generalist embodied agents: Methods and lessons. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10644–10655, June 2025.

V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihan Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, Aohan Zeng, Baoxu Wang, Bin Chen, Boyan Shi, Changyu Pang, Chenhui Zhang, Da Yin, Fan Yang, Guoqing Chen, Jiazheng Xu, Jiale Zhu, Jiali Chen, Jing Chen, Jinhao Chen, Jinghao Lin, Jinjiang Wang, Junjie Chen, Leqi Lei, Letian Gong, Leyi Pan, Mingdao Liu, Mingde Xu, Mingzhi Zhang, Qinkai Zheng, Sheng Yang, Shi Zhong, Shiyu Huang, Shuyuan Zhao, Siyan Xue, Shangqin Tu, Shengbiao Meng, Tianshu Zhang, Tianwei Luo, Tianxiang Hao, Tianyu Tong, Wenkai Li, Wei Jia, Xiao Liu, Xiaohan Zhang, Xin Lyu, Xinyue Fan,

Xuancheng Huang, Yanling Wang, Yadong Xue, Yanfeng Wang, Yanzi Wang, Yifan An, Yifan Du, Yiming Shi, Yiheng Huang, Yilin Niu, Yuan Wang, Yuanchang Yue, Yuchen Li, Yutao Zhang, Yuting Wang, Yu Wang, Yuxuan Zhang, Zhao Xue, Zhenyu Hou, Zhengxiao Du, Zihan Wang, Peng Zhang, Debing Liu, Bin Xu, Juanzi Li, Minlie Huang, Yuxiao Dong, and Jie Tang. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning, 2025. URL https://arxiv.org/abs/2507.01006.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, Jingjing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025. URL https://arxiv.org/abs/2508.18265.

Isadora White, Kolby Nottingham, Ayush Maniar, Max Robinson, Hansen Lillemark, Mehul Maheshwari, Lianhui Qin, and Prithviraj Ammanabrolu. Collaborating action by action: A multi-agent llm framework for embodied reasoning, 2025. URL https://arxiv.org/abs/2504.17950.

Hanqing Yang, Narjes Nourzad, Shiyu Chen, Marie Siew, Jingdi Chen, and Carlee Joe-Wong. Coop$^2$: Defining, observing, and repairing cooperation in llm multi-agent systems, 2026. URL https://arxiv.org/abs/2603.00349.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, Heng Ji, Huan Zhang, and Tong Zhang. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=DgGF2LEBPS.

Yijun Yang, Tianyi Zhou, Kanxue Li, Dapeng Tao, Lusong Li, Li Shen, Xiaodong He, Jing Jiang, and Yuhui Shi. Embodied multi-modal agent trained by an llm from a parallel textworld. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26275–26285, June 2024.

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. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 19373–19401, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/54b8b4e0b4ba4aad112e84f32e3b5dbb-Paper-Conference.pdf.

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. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 49996–50019, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/7d03c6bf9f07acb4038eea96c63db52d-Paper-Conference.pdf.

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

Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, Zhaoye Fei, Zhangyue Yin, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11142–11152, October 2025b.

Qinhong Zhou, Hongxin Zhang, Xiangye Lin, Zheyuan Zhang, Yutian Chen, Wenjun Liu, Zunzhe Zhang, Sunli Chen, Lixing Fang, Qiushi Lyu, Xinyu Sun, Jincheng Yang, Zeyuan Wang, Bao Dang, Zhehuan Chen, Daksha Ladia, Quang Dang, Jiageng Liu, and Chuang Gan. Virtual community: An open world for humans, robots, and society. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 63971–64010, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/68933e3533add841e115a5605c76eeba-Paper-Conference.pdf.

Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong Wang, Cheng Qian, Xiangru Tang, Heng Ji, and Jiaxuan You. MultiAgentBench : Evaluating the collaboration and competition of LLM agents. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8580–8622, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.421. URL https://aclanthology.org/2025.acl-long.421/.

Table 5: Five representative MECo-House task templates. `ON(obj, rec)` and `IN(obj, rec)` denote placing an object on a surface or inside a container. Predicates show the possible goal relations for each template; instance-specific targets and quantities are omitted.

| Task Template | Predicate Set |
|---|---|
| Self-care Routine | ON(barsoap, bathroomcounter),<br>ON(toothbrush, bathroomcounter),<br>ON(toothpaste, bathroomcounter),<br>ON(towel, bathroomcounter) |
| Wash Dishes | IN(cutleryfork, dishwasher),<br>IN(plate, dishwasher),<br>IN(waterglass, dishwasher),<br>IN(wineglass, dishwasher) |
| Store Play Supplies | IN(crayons, cabinet),<br>IN(sportsball, cabinet),<br>IN(toy, cabinet) |
| Outfit Preparation | ON(clothespants, bed),<br>ON(clothesshirt, bed),<br>ON(hanger, bed),<br>ON(slippers, bed) |
| Breakfast Setup | ON(banana, kitchentable),<br>ON(breadslice, kitchentable),<br>ON(cereal, kitchentable),<br>ON(dishbowl, kitchentable),<br>ON(milk, kitchentable) |

# A MECoBench

This section describes the task definitions, construction details, and spatial settings of the two MECoBench domains: MECo-House and MECo-City.

## A.1 MECo-House

### A.1.1 Task Definition

Table 5 shows examples of task templates drawn from the 20 MECo-House task types. Each template defines a pool of object-placement predicates; a task instance selects the required predicates and object counts and grounds them to target receptacles in a concrete scene. Figure 6 illustrates the full goal specification for a Gaming Setup instance.

### A.1.2 Benchmark Construction

The construction of MECo-House tasks follows a two-step pipeline (Figure 7). In step one, task templates are instantiated into concrete scenes with randomized object placement and validated for feasibility.

**Goal Sampling.** For a given task type, we sample subgoals from predefined object-count ranges, specifying the target objects, required counts, and optional room or surface constraints. To mitigate visibility issues such as occlusion, we add 0–2 redundant instances for each target object category during scene instantiation. For large objects that occupy more surface space, such as keyboards, boardgames, and magazines, the redundancy is limited to 0–1 instance.

**Scene Instantiation and Object Placement.** Once a template is sampled, the target objects are assigned to legal initial locations based on the predefined legal placement (Table 6). Objects are placed randomly while respecting constraints:

- No object is placed on the same surfaces or containers that are reserved for other goal locations.

**Find and put 1 magazine on coffeetable.** Possible locations: One magazine is on the cabinet, the cabinet is inside the bedroom. Appearance: Glossy comic-style magazine with orange-and-blue cover, vertical rectangular shape, bold white title text, and a blue robotic hero illustration. coffeetable appearance: Square low coffee table with light gray wood top, dark charcoal recessed base, and a centered four-panel top divided by thin seams.;
**Find and put 1 toy on coffeetable.** Possible locations: One toy is on the bathroomcounter, the bathroomcounter is inside the bathroom. / One toy is on the sofa, the sofa is inside the kitchen. Appearance: Colorful wooden toy truck with blue box body, green cab, yellow base, and red wheels.;
**Find and put 1 book on coffeetable.** Possible locations: One book is on the kitchentable, the kitchentable is inside the kitchen. Appearance: Purple hardcover book with white front cover, Cisco Unity Connection title, gold page edges, rectangular shape.;
**Find and put 2 remotecontrols on coffeetable.** Possible locations: One remotecontrol is on the desk, the desk is inside the bedroom. / One remotecontrol is on the desk, the desk is inside the livingroom. / One remotecontrol is inside the fridge, the fridge is inside the kitchen. Appearance: Gray plastic TV remote, rounded rectangular shape, slightly tapered ends, white button face with black and red buttons.;
**Find and put 1 boardgame on coffeetable.** Possible locations: One boardgame is on the sofa, the sofa is inside the livingroom. / One boardgame is inside the fridge, the fridge is inside the kitchen. Appearance: Square yellow board with dark gray play area, raised center platform, and small red, yellow, green, and blue tokens.;
coffeetable is in livingroom.

Figure 6: Example MECo-House goal specification. The goal specifies target objects, their possible locations highlighted in blue, and appearance descriptions highlighted in orange.

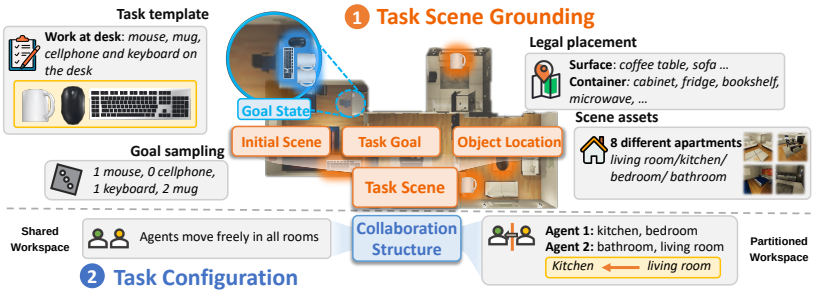



Figure 7: **Data construction pipeline of MECo-House**. Each task is first grounded from a high-level task into a concrete scene, then set collaboration configuration for execution in a shared or partitioned workspace.

- Surface space is checked to ensure sufficient area for the object.
- Containers are verified for available slots using their bounding boxes.

This procedure ensures that the initial scene is diverse yet consistent with the task template.

Table 6: Legal placement locations used for randomized object initialization in MECo-House.

| Type | Legal Locations |
|---|---|
| Container | fridge, stove, dishwasher, microwave, cabinet, bookshelf |
| Surface | sofa, cabinet, bathroomcounter, nightstand, bench, bed, desk, tvstand, kitchencounter, coffeetable, stove, kitchentable |

**Validation and Goal Specification.** After placement, the VirtualHome environment is used to verify scene feasibility:

- The environment graph is updated to reflect the actual object positions.
- Target object counts are checked to ensure that all task objects exist in the scene.
- Predicates from the template are instantiated with the real node IDs in the scene.

- Tasks failing these checks (e.g., insufficient instances or invalid placements) are discarded, and a new attempt is generated.

### A.1.3 BENCHMARK STATISTICS

MECo-House contains 240 base tasks across 20 templates, with 12 instances per template. The dataset includes 55 task-object types represented by 67 distinct 3D object models, 59 distinct 3D furniture models used for initial object placement, and 30 apartments. Each base instance is counted once, irrespective of the number of agents sharing its goals.

Figure 8(a) summarizes the 20 task templates, their associated object types, and semantic categories. Difficulty is defined by the total number of required objects, including repeated instances of the same type. Levels L1–L6 correspond to 2–7 objects and contain 40, 41, 40, 40, 42, and 37 instances, respectively. Their proportions among all 240 base tasks are shown in Figure 8(b).

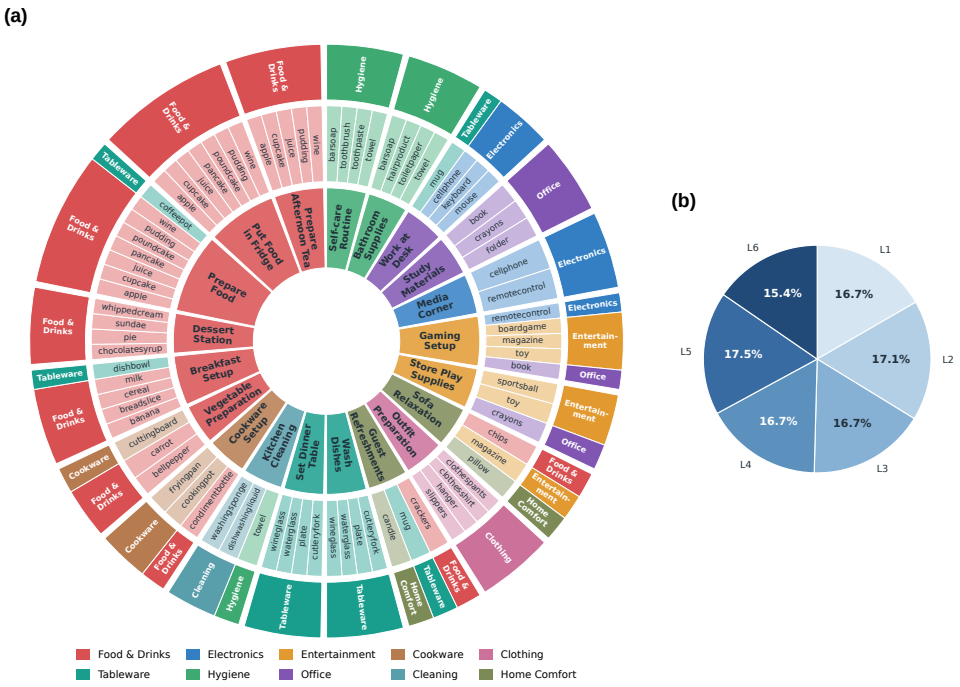


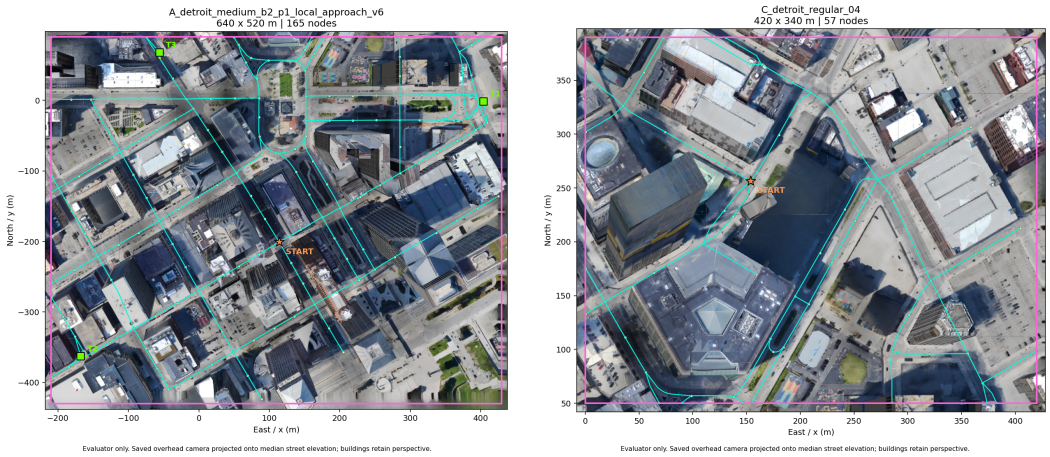

Figure 8: **MECo-House task composition and difficulty.** (a) The inner, middle, and outer rings show 20 task templates, associated object types, and semantic categories, respectively. (b) Distribution of 240 base tasks across six difficulty levels, where L1–L6 require 2–7 objects.

## A.2 MECO-CITY

MECo-City includes MECo-Nav and MECo-Exp in urban environments from Virtual-Community Zhou et al. (2026). Figure 9 shows representative task layouts.

**MECo-Nav.** Small maps span $520\,\mathrm{m} \times 420\,\mathrm{m}$ with two target cars, and large maps span $640\,\mathrm{m} \times 520\,\mathrm{m}$ with three. In the **Shared Workspace**, agents search for and visually confirm target cars. In the **Partitioned Workspace**, agents operate in assigned zones, collect parcels from specified vehicles, coordinate transfers at handover sites, and deliver the parcels to destination vehicles.

**MECo-Exp.** Agents explore a $420\,\mathrm{m} \times 340\,\mathrm{m}$ map in the Shared Workspace until team coverage reaches $0.7$. Coverage is the union of areas observed by the team, using a $60\,\mathrm{m}$ visibility radius at each position and excluding building-occluded areas.




(a) MECo-Nav (Shared Workspace). (b) MECo-Exp: urban exploration.

Figure 9: **Example MECo-City task layouts.** Cyan lines show navigation graphs, magenta outlines mark task boundaries, and orange stars indicate starting positions. The MECo-Nav example contains three target cars (green markers). These overhead views are for illustration; agent inputs are shown in Figure 12.

# B EVALUATION

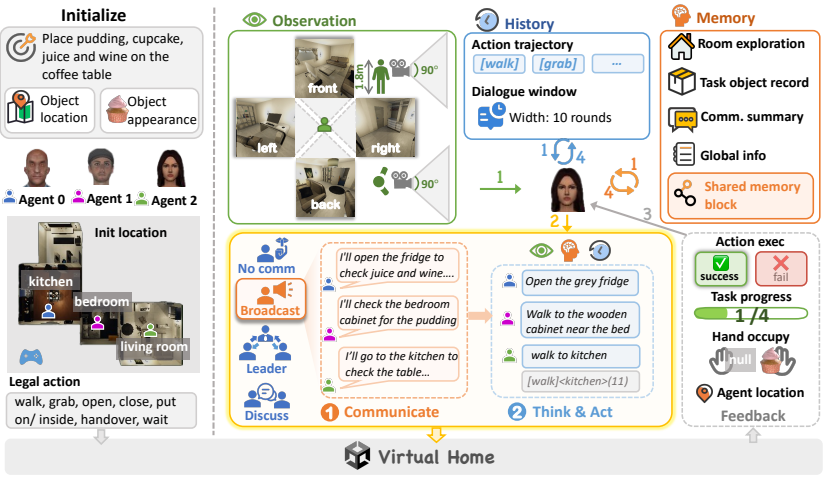



Figure 10: **Evaluation workflow illustrated with MECo-House.** Agents receive task goals and prior information, perceive the environment, communicate under different protocols, reason with history and memory, and execute actions with feedback.

## B.1 EVALUATION SETUP

### B.1.1 MECO-HOUSE OBSERVATIONS

Each MECo-House observation view is rendered at $256 \times 256$ pixels, resulting in a four-view concatenated observation of $256 \times 1024$. The four views are arranged in the order: front, back, left, and right. Both horizontal and vertical fields of view are set to $90^\circ$. The agent model height is 1.8 m, and the camera is mounted at the same height with a forward offset of 0.15 m relative to the agent's position. The camera is pitched downward by $30^\circ$ to capture more of the nearby surfaces and objects. Figure 11 shows an example of the resulting concatenated observation.



Figure 11: Example four-view observation in MECo-House.

### B.1.2 MECo-City Observations and Movement

Agents receive directional street views and a north-up map recording explored areas, agent positions, and labeled movement options (Figure 12); unobserved areas remain unknown. Each `move` follows a selected road option for up to $40\,\mathrm{m}$ in MECo-Nav or $65\,\mathrm{m}$ in MECo-Exp, stopping earlier at an intersection.




(a) Directional street views.

(b) Observation-history map.

Figure 12: **Example MECo-City agent inputs.** (a) North, east, south, and west street views with movement choices. (b) A north-up observation-history map for Shared Workspace MECo-Nav; white areas are unknown.

### B.1.3 Action Budgets

For the Shared Workspace, we compare single-agent and multi-agent execution under the same total action budget $B$: 60 steps for MECo-House, 80 for MECo-Nav, and 42 for MECo-Exp. For an $n$-agent team, the budget is divided evenly across agents, giving each agent $B/n$ steps. The Partitioned Workspace applies only to MECo-House and MECo-Nav and requires two-agent collaboration, with total budgets of 80 and 120 steps, respectively.

## B.2 Model Versions

Table 7 summarizes the model versions, full identifiers, sizes, reasoning capabilities, and generation settings used in our experiments. Proprietary models are accessed through their official APIs, while open-source models are locally deployed using vLLM Kwon et al. (2023).

For all evaluated models, we adopt the recommended generation configurations whenever available. For closed-source models, we use the official APIs with their default generation settings. For open-source models, we follow the generation configurations specified in the corresponding model cards or vLLM Kwon et al. (2023) documentation to ensure strong and reliable performance.

Table 7: Model versions, sizes, reasoning capabilities, and generation settings used in our evaluation.

| Model Name | Full Name | Size | Reasoning Capability | Temperature | Top-p | Top-k |
|---|---|---|---|---|---|---|
| GPT-5-mini | `gpt-5-mini` | Unknown | Reasoning enabled; medium effort | 1.0 | 1.0 | - |
| GPT-5.4 | `gpt-5.4` | Unknown | Reasoning enabled; medium effort | 1.0 | 0.98 | - |
| Gemini-3.1-Pro | `gemini-3.1-pro-preview` | Unknown | Reasoning enabled; medium effort | 1.0 | 0.95 | - |
| Qwen3-8B-VL | `Qwen/Qwen3-VL-8B-Instruct` | 8B | No explicit reasoning | 0.7 | 0.8 | 20 |
| Qwen3-32B-VL | `Qwen/Qwen3-VL-32B-Instruct` | 32B | No explicit reasoning | 0.7 | 0.8 | 20 |
| Qwen3-235B-VL | `Qwen/Qwen3-VL-235B-A22B-Instruct` | 235B total (22B activated) | No explicit reasoning | 0.7 | 0.8 | 20 |
| Qwen3.5-9B | `Qwen/Qwen3.5-9B` | 9B | Reasoning enabled | 1.0 | 0.95 | 20 |
| Qwen3.5-27B | `Qwen/Qwen3.5-27B` | 27B | Reasoning enabled | 1.0 | 0.95 | 20 |
| Gemma4-26B | `google/gemma-4-26b-it` | 26B total (4B activated) | Reasoning enabled | 1.0 | 0.95 | 64 |
| Gemma4-31B | `google/gemma-4-31b-it` | 31B | Reasoning enabled | 1.0 | 0.95 | 64 |
| InternVL-3.5-38B | `OpenGVLab/InternVL3_5-38B` | 38B | No explicit reasoning | 0.6 | 0.95 | - |
| InternVL-3.5-241B | `OpenGVLab/InternVL3_5-241B-A28B` | 241B total (28B activated) | No explicit reasoning | 0.6 | 0.95 | - |
| Llama-4 Scout | `meta-llama/Llama-4-Scout-17B-16E-Instruct` | 109B total (17B activated) | No explicit reasoning | 0.6 | 0.9 | - |
| GLM-4.6V | `zai-org/GLM-4.6V` | 106B total (12B activated) | Reasoning enabled | 0.8 | 0.6 | 2 |
| GLM-4.6V-Flash | `zai-org/GLM-4.6V-Flash` | 9B | Reasoning enabled | 0.8 | 0.6 | 2 |

## B.3 Action Space

### B.3.1 MECo-House Actions

Table 8: Action space for agents in MECo-House. Shared Workspace tasks include basic navigation and object manipulation actions. Partitioned Workspace tasks introduce cross-room handover actions.

| Action | Description | Parameters |
|---|---|---|
| | *Shared Workspace Actions* | |
| `walk` | Move towards a visible object, furniture, or door | object |
| `walk_to_room` | Move directly to another room | bedroom / livingroom / kitchen / bathroom |
| `grab` | Pick up a visible and grabbable object | object |
| `open` | Open a container | container |
| `put_on` | Place the held object on a visible surface | object (holding), surface |
| `put_in` | Place the held object inside a visible container | object (holding), container |
| | *Additional Partitioned Workspace Actions* | |
| `walk_to_door` | Move to a specific door in the environment topology | door_id |
| `handover` | Give a held object to another agent when close (<2m) | object(holding), recipient agent name |
| `receive` | Wait to receive an object | none |

Table 8 lists the MECo-House action space under both workspace settings. Shared Workspace tasks include basic navigation and object manipulation actions, while partitioned workspace tasks introduce cross-room handover actions.

**Single-Agent Action Legality** Each MECo-House agent action must satisfy the following legality constraints:

- **Hand occupancy:** Each agent can hold at most two objects. When both hands are occupied, the agent cannot execute `grab` or `open`. For `put_*` and `handover`, the object to be placed or handed over must already be held by the acting agent.
- **Visibility:** The target objects of `walk`, `grab`, and `put_*` actions must be visible in the agent's current observation.
- **Container state:** An agent cannot execute `open` on a container that is already open.
- **Room accessibility:** For `walk_*`, `grab`, and `put_*` actions, the target must be located in one of the rooms accessible to the agent.

- **Invalid actions:** Any action name not listed in Table 8 is considered invalid.

**Multi-Agent Interaction Constraints** When multiple MECo-House agents act simultaneously, additional interaction constraints are applied:

- **Conflict resolution:** The conflict set includes `grab` and `open`. If multiple agents attempt the same conflict action on the same object at the same time, one agent is randomly selected as the *winner*, while the others remain idle.
- **Handover distance:** A `handover` action is valid only when the giver and receiver are within 2 meters of each other.

### B.3.2 MECo-City Actions

MECo-City uses discrete actions: each agent outputs one JSON action per round. Table 9 lists the actions and their applicability to each workspace setting.

Table 9: **MECo-City action space.** Nav-S and Nav-P denote MECo-Nav in Shared and Partitioned Workspace, respectively; Exp denotes MECo-Exp (Shared Workspace).

| Action | Example Parameters | Meaning | Tasks |
|---|---|---|---|
| `move` | `"choice":"A"` | Follow the selected road option to an adjacent node. | Nav-S, Nav-P, Exp |
| `found` | `"candidate":"M1"` | Visually confirm the current vehicle candidate. | Nav-S, Nav-P |
| `pickup` | `"item":"P1", "candidate":"M1"` | Collect the specified parcel from a vehicle. | Nav-P |
| `handover` | `"item":"P1", "to_agent":1, "site":"H1"` | Transfer the parcel to another agent at a handover site. | Nav-P |
| `receive` | None required | Wait to receive a parcel. | Nav-P |
| `deliver` | `"item":"P1", "candidate":"M1"` | Deliver the parcel to the specified vehicle. | Nav-P |
| `wait` | None | Remain in place for one round. | Nav-S, Nav-P, Exp |
| `stop` | None | End this agent's subsequent actions. | Nav-S, Nav-P, Exp |

### B.4 MECo-House Action Decoding and Grounding

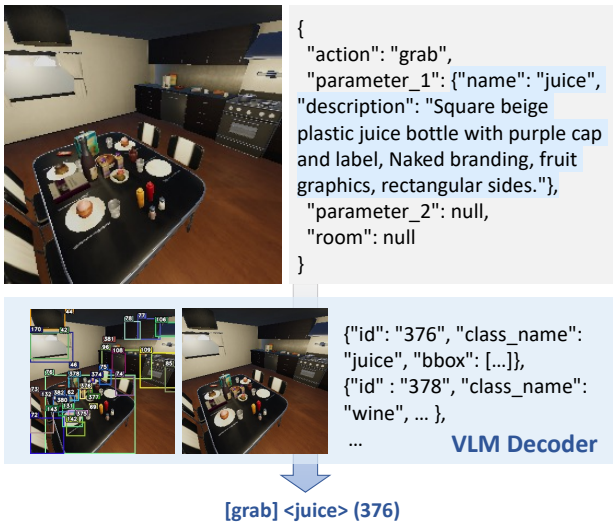



Figure 13: Example of action decoding and grounding in MECo-House.

MECo-House agents output semantic actions based on their observation rather than executable simulator commands. Thus, each generated action is decoded and grounded before execution.

Actions that do not require object grounding, such as `receive`, `walk_to_room/door`, and `handover`, are resolved by rule-based mappings. For object-centric actions, including `walk`, `grab`, `open`, `put_on`, and `put_in`, semantic object descriptions are grounded to concrete simulator object identifiers.

As shown in Fig. 13, the top panels present the agent's first-person observation and generated semantic action. The decoder then uses visible object candidates, their bounding boxes, and metadata, together with both the original and bbox-annotated views, to select the target simulator ID and produce the final executable action.

# C Core Prompt Templates

We present the core communication and action prompts used by MECoBench agents.

## C.1 Communicate Phase

Figures 14–17 present the broadcast, discussion, and leader-based communication prompts.

```
You are an embodied agent in a 3D simulated environment.
At this stage, based on the current observation, memory, history, and task goal, you must decide whether to send ONE helpful message to other agents.

## Communication
Send a message if it is helpful for task progress and remain silent if the message would be redundant, speculative, or misleading.
Your goal is to complete the task with the **smallest number of action steps**. Therefore, avoid waiting or letting others wait. We should all actively engage and search for tasks to contribute to.
The valid actions we can take are: $VALID_ACTIONS$. Each agent can only perform one action at a time, and can hold at most two objects at a time. \n
### MESSAGE Output Format
- If silent, output exactly:
<NO_MESSAGE>
Otherwise, output the message text ONLY.

## Output Format
[THINKING]
Brief reasoning for the message decision.
[MESSAGE]
<NO_MESSAGE> or message text

## Inputs
$INPUT_BLOCK$
```

Figure 14: Broadcast protocol communication prompt. This module asks each agent to decide whether to send a concise, useful message before physical action planning.

```
You are an embodied agent in a 3D simulated environment.
At this stage, you must send ONE report message to the leader agent: $LEADER_NAME$ and wait for the leader's assignment.
Base the report on your current observation, memory, dialogue, and task goal.

## Report
Report only task-relevant observations, status, useful feedback and progress to the leader (Agent 0).
If any action fails, you should report the failure and possible reason to the leader.\n
MESSAGE rules:
- If you have nothing useful to report, output exactly <NO_MESSAGE>.
- Otherwise output one concise report message to the leader.
- Do not invent visual facts.

## Output Format
[THINKING]
Brief reasoning for the report.
[MESSAGE]
<NO_MESSAGE> or message text
```

Figure 15: Centralized mode: worker report prompt.

You are an embodied agent in a 3D simulated environment.
This is the FIRST discussion round for the current step.
Based on the current observation, memory, dialogue, and task goal, decide whether more discussion is needed before acting.

You are an embodied agent in a 3D simulated environment.
This is a FOLLOW-UP text-only discussion round for the current step.
Use the dialogue so far and your private local context from the first round to decide whether more discussion is needed before acting.

## Discussion
You may send ONE helpful discussion message to the other agents.
Your goal is to complete the task with the **smallest number of action steps**. Therefore, avoid waiting idly or letting others wait. We should all actively engage and search for tasks to contribute to.
The valid actions we can take are: $VALID_ACTIONS$. Each agent can only perform one action at a time, and can hold at most two objects at a time.\n
READY rules:
- Output <READY> if, from your perspective, no more discussion is needed before agents take actions.
- Output <CONTINUE> if more discussion is still needed.\n
MESSAGE rules:
- If you have nothing useful to add, output exactly <NO_MESSAGE>.
- Otherwise output one concise, task-relevant message.
- Do not fabricate visual facts.

## Output Format
[THINKING]
Brief reasoning for the discussion decision.
[READY]
<READY> or <CONTINUE>
[MESSAGE]
<NO_MESSAGE> or message text
[LOCAL_CONTEXT]
A short private visual grounding note for your own later text-only discussion rounds.

## Inputs
$INPUT_BLOCK$

Figure 16: Discussion protocol prompt. The first round prompt is highlighted in blue, and the follow-up rounds prompt is highlighted in orange.

You are an embodied agent in a 3D simulated environment.
At this stage, you are acting as the leader for this step.
Based on your current observation, memory, dialogue, and the workers' reports,
assign exactly ONE next-step instruction to each agent, including yourself.
You have access to a combined image that shows every agent's current observation views (your own view on top, followed by each worker's view, each row labeled with the agent name).
Together with your memory, dialogue, and the workers' text reports, use this shared visual information to assign exactly ONE next-step instruction to each agent, including yourself.

## Assignment
You must output exactly one assignment for each agent in this list:
$AGENT_NAMES$
Your goal is to complete the task with the **smallest number of action steps**. Therefore, avoid letting any agent stay idle or wait for other agents.\n
Rules:
- Use the exact agent names from the list.
- Set "type" to "assignment" for every item.
- Give concise, actionable instructions for THIS step only.
- The valid actions we can take are: $VALID_ACTIONS$. Each agent can only perform one action at a time, and can hold at most two objects at a time. Get full use of all agents' actions and hands to minimize the total number of action steps.
- Do not assign an agent to do multiple unrelated things in one message.
- If the worker reports failure, you should assign a different action. Do not let the worker retry the failed action if the environment has not changed.

## Output Format
[THINKING]
Brief reasoning for the assignments.
[MESSAGES]
A JSON array like:
[
 {"receiver": "<exact agent name>", "type": "assignment", "content": "..."}
]

Figure 17: Centralized mode: leader assignment prompt. Text-only communication is highlighted in blue, and the vision-augmented is highlighted in orange.

### C.2 ACT PHASE

Figures 18 and 19 present the action-selection template and its physical interaction rules and output format.

```
You are an embodied agent in a 3D simulated environment.
At this stage, based on the current observation, memory, current communication context, and task goal,
you must produce TWO outputs in order:
1) Decide the NEXT single physical action.
2) Update the memory.
You coordinate with other agents ONLY through a shared memory object. There is no current-round dialogue in
this mode.
At this stage, based on the current observation, shared memory, previous action feedback, and task goal,
you must produce TWO outputs in order:
1) Decide the NEXT single physical action.
2) Update the shared memory with observed facts, your current plan, and your own status.

## Physical Actions
$ACTION_RULES$

## Memory Update
$MEMORY_RULES$

## Output Format
[THINKING]
Brief reasoning for the next action.
[ACTION]
The JSON action object. Only one action block is allowed.
[THINKING]
Brief reasoning for the memory update.
[MEMORY]
The updated memory JSON.
```

Figure 18: Action and memory update prompt. Normal memory is highlighted in blue and shared memory is highlighted in orange.

```
## Physical Actions
### Valid Actions
- put one object you are holding on a surface or into a container
- open an enclosure
- grab an object
- walk to something visible
- walk to another room (allowed rooms:$ALLOWED_ROOMS$)\n
### Interaction Rules
- When an action involves any object, specify the name and a clear visual/location description.
- You may interact (walk to, grab, put, open) with objects ONLY when they are currently visible.
- If the object is not visible, first walk to the room includes it. If still occluded, walk to a nearby visible landmark
(e.g., table/counter) until the target becomes visible.
- You can hold at most TWO objects at the same time. If both hands are full, put one object down before
grabbing another object or opening an enclosure.
- Do NOT repeat a failed action unless the environment has changed. Do NOT repeating the same action
consecutively.
- You may walk directly to any room using the room name except the current one.\n
## ACTION Output Format
Output a single JSON object, with no extra text:
{
  "action": "<one of: walk / walk_to_room / grab / put_in / put_on / open >",
  "parameter_1": {"name": "<class name>", "description": "<short description>"} | null,
  "parameter_2": {"name": "<class name>", "description": "<short description>"} | null,
  "room": "<bedroom|livingroom|kitchen|bathroom>" | null
}\n
Parameter rules:
- walk/open/grab: use "parameter_1" only.
- put_in/put_on: "parameter_1" = held object, "parameter_2" = container/surface.
- walk_to_room: set "room" and set both parameters to null.
- Actions with no parameters: set "parameter_1", "parameter_2", and "room" to null.
```

Figure 19: Action rules block.

# D EXPERIMENT RESULTS

In this section, we provide detailed results for the experiments in Sections 4.2, 4.3, and 5 and Appendix G, as well as some additional experimental results.

## D.1 MODEL PERFORMANCE ACROSS WORKSPACE SETTINGS

Table 2 in the main text reports individual and collaborative performance across models. Overall, GPT-5.4 achieves the best performance. GPT-5-mini and Gemma4-31B are the second-tier models, showing consistently strong results across the two collaboration structures. In addition, Qwen3.5, Gemma4-26B, and Gemini-3.1-Pro also exhibit relatively strong performance. Qwen3-VL-32B and Qwen3-VL-235B fall into the middle-performance group, with relatively balanced results across shared workspace and partitioned workspace tasks. Qwen3-VL-8B is a relatively weak model, but its performance remains comparatively balanced between the two coordination structures. In contrast, InternVL-3.5 and GLM-4.6V struggle particularly on partitioned workspace tasks, where precise collaboration is required. Finally, Llama-4-Scout and GLM-4.6V-Flash perform poorly under both collaboration structures.

**Model scaling effects.** Overall, except for InternVL-3.5, larger models generally perform better within the same family. However, scaling tends to amplify existing task-specific patterns rather than change them. Some families remain balanced across shared workspace and partitioned workspace tasks, while others are consistently stronger in single-agent or shared workspace settings, or particularly weak on partitioned workspace tasks requiring precise coordination. For example, Gemma4 benefits from scaling especially on partitioned workspace tasks, whereas InternVL-3.5 achieves stronger single-agent shared workspace performance at larger scale but degrades in partitioned workspace collaboration. This suggests that stronger individual capability does not necessarily translate into better communication and coordination.

Based on model performance and cost-effectiveness, we select Qwen3-VL-32B and Qwen3-VL-8B as two representative models for studying team-size scaling and related factors in subsequent experiments. For strong models, we conduct most experiments with GPT-5-mini and Gemma4-31B, and include GPT-5.4 when necessary.

Table 10: Team-size scaling in a shared workspace, with broadcast communication or no communication. Best results within each model and protocol are bold.

| #Ag. | With comm. | | | Without Comm. | | |
|---|---|---|---|---|---|---|
| | SR | CR | AUC | SR | CR | AUC |
| *Qwen3-VL-8B* | | | | | | |
| 1 | 0.354 | 0.620 | 0.419 | 0.354 | 0.620 | **0.419** |
| 2 | 0.458 | 0.770 | 0.535 | 0.271 | 0.546 | 0.339 |
| 3 | **0.521** | **0.837** | **0.558** | **0.396** | **0.655** | 0.411 |
| 4 | 0.469 | 0.794 | 0.501 | 0.323 | 0.642 | 0.409 |
| 5 | 0.458 | 0.794 | 0.468 | – | – | – |
| *Qwen3-VL-32B* | | | | | | |
| 1 | 0.583 | 0.753 | 0.542 | **0.583** | 0.753 | 0.542 |
| 2 | 0.802 | 0.934 | **0.694** | **0.583** | **0.802** | **0.567** |
| 3 | 0.833 | 0.943 | 0.656 | 0.500 | 0.705 | 0.481 |
| 4 | **0.844** | **0.954** | 0.658 | 0.469 | 0.720 | 0.468 |
| 5 | 0.740 | 0.892 | 0.567 | – | – | – |

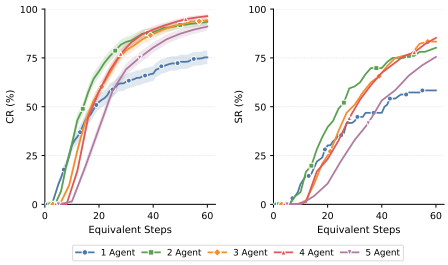



Figure 20: Task progress with one to five Qwen3-VL-32B agents in a shared workspace under broadcast communication.

**Team size scaling effects.** Table 10 shows the detailed result for Figure 4(a) and 3(b). The results show that: (1) Simply increasing the number of agents does not necessarily improve performance.

Without communication, adding more agents can even introduce disorder and redundant actions, leading to lower efficiency and worse task completion. However, when agents are allowed to communicate, multi-agent collaboration becomes more effective in coordinating actions and completing tasks. (2) More agents are not always better. Qwen3-VL-32B benefits from additional agents up to four agents, while Qwen3-VL-8B reaches its best performance with three agents and drops afterward. This suggests that weaker models are more affected by coordination overhead. (3) Collaboration can partially compensate for limited individual model capability, but cannot fully remove the capability gap. With communication, three Qwen3-VL-8B agents approach the single-agent performance of Qwen3-VL-32B, but the final performance remains constrained by the underlying model capability. Table 11 shows the detailed results in figure 4(b). This indicate that multi-agent collaboration is especially helpful for more complex tasks with more target objects. For 7-object tasks, the SR nearly doubles for both models: from 0.385 to 0.750 for Qwen3-VL-32B, and from 0.077 to 0.154 for Qwen3-VL-8B. A task progress comprison of different team size shown in Figure 20.

Table 11: Success rate and completion rate by number of objects for Qwen3-VL models on Shared Workspace tasks. Best results for each model are shown in bold, and second-best results are underlined.

| Model | #Objects | 1 Agent | | 2 Agents | | 3 Agents | | 4 Agents | | 5 Agents | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | SR | CR | SR | CR | SR | CR | SR | CR | SR | CR |
| Qwen3-32B-VL | 2 | 0.875 | 0.938 | 0.875 | 0.938 | 0.938 | 0.969 | 0.938 | 0.969 | **1.000** | **1.000** |
| | 3 | 0.765 | 0.863 | **1.000** | **1.000** | **1.000** | **1.000** | 0.941 | 0.980 | 0.875 | 0.938 |
| | 4 | 0.625 | 0.812 | 0.938 | 0.984 | **1.000** | **1.000** | **1.000** | **1.000** | 0.750 | 0.922 |
| | 5 | 0.438 | 0.650 | 0.688 | 0.838 | 0.688 | 0.888 | **0.812** | **0.938** | 0.800 | 0.867 |
| | 6 | 0.389 | 0.630 | 0.667 | 0.935 | **0.778** | **0.944** | 0.667 | 0.935 | 0.500 | 0.880 |
| | 7 | 0.385 | 0.604 | 0.615 | 0.901 | 0.583 | 0.905 | **0.750** | **0.964** | 0.667 | 0.917 |
| Qwen3-8B-VL | 2 | 0.750 | 0.875 | 0.875 | 0.938 | **0.938** | **0.969** | 0.812 | 0.906 | 0.875 | 0.938 |
| | 3 | 0.588 | 0.784 | 0.824 | 0.922 | **0.882** | **0.941** | 0.824 | 0.922 | 0.706 | 0.863 |
| | 4 | 0.312 | 0.562 | **0.500** | **0.828** | **0.500** | **0.828** | **0.500** | **0.828** | 0.438 | 0.766 |
| | 5 | 0.250 | 0.562 | 0.375 | 0.713 | 0.438 | 0.800 | 0.333 | 0.747 | **0.500** | **0.787** |
| | 6 | 0.111 | 0.444 | 0.056 | 0.685 | 0.167 | 0.713 | **0.278** | **0.741** | 0.111 | 0.704 |
| | 7 | 0.077 | 0.472 | 0.077 | 0.483 | **0.154** | **0.769** | 0.000 | 0.637 | 0.077 | 0.692 |

### D.2 Role of communication

Table 12 shows the detailed results in figure 3(a) when ablating communication from collaboration. Removing communication degrades both task performance and collaboration quality. In shared workspace tasks, it generally increases CAR, indicating more conflicts actions between agents. In partitioned workspace tasks, the impact is more severe, as agents fail to coordinate cross-region handovers, causing HFR to rise substantially.

Table 12: Results for 2-agent broadcast and no-communication protocols on Shared Workspace and Partitioned Workspace tasks. NA indicates that no handover attempt was made.

| Model | Mode | Shared Workspace | | | | Partitioned Workspace | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | SR | CR | AUC | CAR | SR | CR | AUC | HFR |
| Gemma4-31B | Broadcast | 0.833 | 0.945 | 0.689 | 0.019 | 0.823 | 0.904 | 0.620 | 0.322 |
| | No Comm | 0.719 | 0.863 | 0.558 | 0.023 | 0.177 | 0.464 | 0.308 | 0.486 |
| Qwen3-32B-VL | Broadcast | 0.802 | 0.934 | 0.694 | 0.020 | 0.365 | 0.620 | 0.393 | 0.294 |
| | No Comm | 0.583 | 0.802 | 0.567 | 0.028 | 0.073 | 0.325 | 0.232 | 0.733 |
| Qwen3.5-9B | Broadcast | 0.938 | 0.985 | 0.764 | 0.008 | 0.615 | 0.844 | 0.558 | 0.400 |
| | No Comm | 0.865 | 0.939 | 0.680 | 0.036 | 0.062 | 0.325 | 0.231 | 0.903 |
| Qwen3-8B-VL | Broadcast | 0.458 | 0.770 | 0.535 | 0.010 | 0.073 | 0.332 | 0.208 | 0.662 |
| | No Comm | 0.274 | 0.552 | 0.342 | 0.017 | 0.042 | 0.182 | 0.132 | NA |

### D.3 Comparison of Different Collaboration Modes

Table 13 gives the detailed results of Figure 4(c). The results show that centralized collaboration generally improves division of labor and task efficiency, reflected by higher DOL and AUC. In

Table 13: Results of different communication protocol under shared workspace and partitioned workspace tasks across models. *Spatial Err* denotes the frequency of spatial errors, where an agent attempts to move to a location outside its assigned region. Best results within each model are shown in bold, and cross-model best results are highlighted in yellow.

| Model | Mode | Shared Workspace | | | | | | Partitioned Workspace | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | SR | CR | AUC | DOL | Tokens | CAR | SR | CR | AUC | Spatial Err | Tokens | HFR |
| ***Closed-source MLLMs*** | | | | | | | | | | | | | |
| GPT-5.4 | Broadcast | **0.900** | **0.961** | 0.700 | 0.773 | **4331.9** | 0.015 | **0.767** | 0.904 | 0.640 | **0.017** | **7328.1** | 0.228 |
| | Discuss | 0.767 | 0.932 | 0.730 | 0.717 | 6266.6 | **0.001** | **0.767** | **0.911** | **0.652** | 0.021 | 7619.3 | **0.177** |
| | Leader | 0.784 | 0.934 | **0.752** | **0.883** | 6558.6 | 0.009 | **0.767** | 0.870 | 0.639 | 0.023 | 8085.5 | 0.223 |
| ***Open-source MLLMs*** | | | | | | | | | | | | | |
| Gemma4-31B | Broadcast | 0.833 | 0.945 | 0.689 | 0.794 | 5381.8 | 0.019 | 0.823 | 0.904 | 0.620 | 0.038 | 8042.4 | 0.322 |
| | Discuss | 0.802 | 0.855 | 0.638 | 0.731 | 6852.5 | **0.001** | 0.812 | 0.922 | 0.625 | 0.025 | 8978.3 | 0.224 |
| | Leader | **0.865** | **0.964** | **0.736** | **0.840** | 6863.0 | 0.003 | **0.854** | **0.963** | **0.652** | **0.016** | 8723.5 | **0.138** |
| | No Comm | 0.719 | 0.863 | 0.558 | 0.706 | **3577.4** | 0.023 | 0.177 | 0.464 | 0.308 | 0.297 | **4539.7** | 0.486 |
| Qwen3-32B-VL | Broadcast | 0.802 | 0.934 | 0.694 | 0.802 | 4682.0 | 0.020 | 0.365 | 0.620 | 0.393 | 0.166 | 6749.4 | **0.294** |
| | Discuss | 0.808 | 0.929 | 0.694 | 0.818 | 6351.7 | **0.003** | 0.333 | 0.583 | 0.374 | 0.100 | 8324.3 | 0.304 |
| | Leader | **0.833** | **0.938** | **0.706** | **0.855** | 5964.7 | 0.012 | **0.396** | **0.622** | **0.408** | **0.046** | 7516.5 | 0.378 |
| | No Comm | 0.583 | 0.802 | 0.567 | 0.629 | **3017.7** | 0.028 | 0.073 | 0.325 | 0.232 | 0.451 | **3559.0** | 0.733 |
| Qwen3-8B-VL | Broadcast | 0.458 | 0.770 | 0.535 | 0.676 | 4585.4 | 0.010 | **0.073** | **0.332** | 0.207 | 0.312 | 6788.2 | 0.662 |
| | Discuss | 0.458 | 0.783 | 0.494 | **0.702** | 7206.1 | 0.011 | 0.021 | 0.283 | 0.195 | 0.319 | 8934.5 | **0.601** |
| | Leader | **0.490** | **0.785** | **0.548** | 0.642 | 5701.5 | **0.005** | 0.052 | **0.332** | **0.227** | **0.223** | 7161.4 | 0.658 |
| | No Comm | 0.274 | 0.552 | 0.342 | 0.357 | **3310.1** | 0.017 | 0.042 | 0.182 | 0.132 | 0.613 | **3475.1** | NA |

contrast, the discuss protocol is more effective at reducing conflicts, leading to lower CAR. For weaker models, centralized collaboration also helps reduce spatial cognition errors in partitioned workspace tasks, suggesting that leader-based coordination can provide useful guidance when agents have limited planning or spatial reasoning ability. Figure 23 shows the comparison of task progress under single agent and different communication protocol in two-agent team.

**Shared Memory.** Figure 21 reports the absolute values of success rate, AUC, and token consumption for broadcast and shared memory, providing a more detailed version of Figure 5(a).

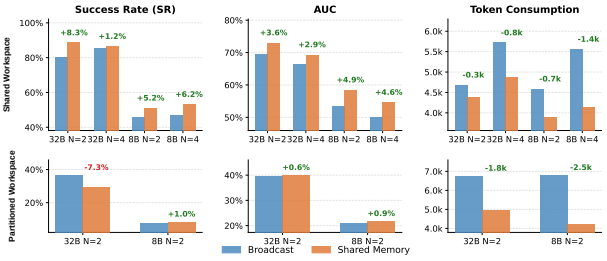



Figure 21: Comparison of explicit broadcast and implicit shared memory protocol. The models are from the Qwen3-VL family.

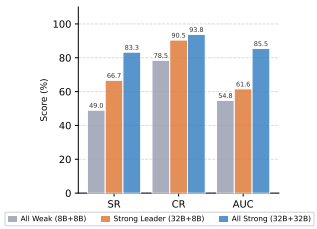



Figure 22: Effect of leader capability in centralized collaboration. The models are from the Qwen3-VL family.

### D.4 Mix Team: Strong Model as Leader.

We further investigate the role of leader in centralized mode. As shown in Figure 22, using a stronger leader improves the mixed team, but does not make it approach the all-strong setting. The mixed team nearly matches the all-strong team in CR, but its SR is only slightly above the average of the all-weak and all-strong teams, and its AUC is even below this average. This indicates that a strong leader mainly improves task coverage, but efficient and overall success still depends heavily on the worker's capability.

### D.5 Performance Under Imperfect Information.

**Missing location priors** Table 14 reports detailed results under the no-prior-location setting on shared workspace tasks across different models. Figure 24 shows the team size scaling effect under stronger exploration demands, where prior location information is removed. Compared with the

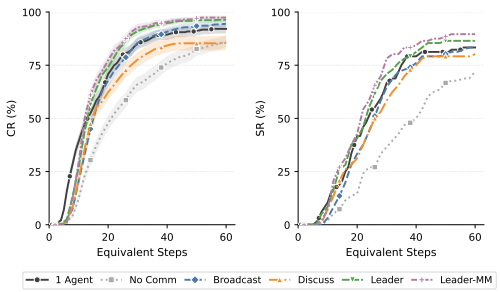



Figure 23: Task progress over equivalent steps under different communication protocols on shared workspace tasks. Results are obtained with Gemma4-31B. Leader-MM denotes the vision-augmented leadership protocol.

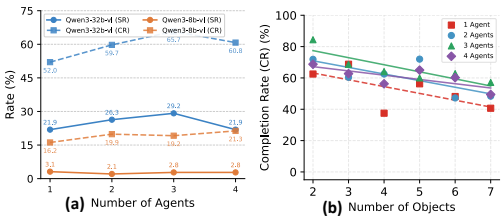



Figure 24: Team-size scaling without location priors. (a) SR and CR versus team size. (b) CR versus the number of target objects, with fitted trends for different team sizes.

full-information setting, Qwen3-32B-VL exhibits a similar but weaker inverted-U-shaped trend, suggesting that moderate team scaling remains helpful. For Qwen3-8B-VL, SR remains low across team sizes, indicating a capability bottleneck. Moreover, when grouping cases by object count, larger teams still show a milder performance drop. More detailed results in Table 16.

Table 14: Results on shared workspace tasks without location information. Each cell reports SR/CR/AUC. Qwen-* is short for Qwen3-VL, Inter-* is short for InterVL3.5.

| **Model** | **1-Agent** | **2-Agent** |
|---|---|---|
| *Closed-source MLLMs* | | |
| GPT-5-mini | **0.323 / 0.630 / 0.436** | **0.344 / 0.691 / 0.436** |
| *Open-source MLLMs* | | |
| Qwen-8B | 0.031 / 0.162 / 0.103 | 0.021 / 0.199 / 0.101 |
| Qwen-32B | 0.219 / 0.520 / 0.344 | 0.260 / 0.591 / 0.365 |
| Qwen-235B | 0.115 / 0.389 / 0.264 | 0.271 / 0.581 / 0.332 |
| Qwen3.5-9B | 0.292 / 0.582 / 0.389 | 0.292 / 0.649 / 0.435 |
| Gemma-31B | 0.240 / 0.561 / 0.358 | 0.229 / 0.589 / 0.347 |
| Intern-38B | 0.010 / 0.146 / 0.087 | 0.073 / 0.253 / 0.144 |
| Intern-241B | 0.021 / 0.201 / 0.113 | 0.062 / 0.217 / 0.114 |
| Llama-4 | 0.021 / 0.089 / 0.058 | 0.000 / 0.107 / 0.056 |

**Ablation on prior information** To investigate how different types of prior information affect agent performance, we conduct an ablation study by removing object appearance descriptions and prior location information. As shown in figure 26(a), prior location information is critical for task completion, as it helps agents quickly locate target objects and reduces inefficient exploration. Removing location information leads to a substantial performance drop across all models. In contrast, the effect of appearance descriptions is model-dependent: GPT-5-mini benefits from appearance cues, Gemma4-31B shows only minor changes, and Qwen3-32B-VL performs worse with appearance-only priors than with no prior. A possible reason is that models with weaker fine-grained visual grounding are more susceptible to misleading appearance descriptions and may confuse visually similar objects. Detailed results are reported in Table 15.

**Noisy location priors.** Table 17 reports detailed results on shared workspace tasks where 30% of the prior location information is noisy, comparing single-agent and two-agent settings.

### D.6 Additional Performance Analysis

Figure 25 reports performance by the number of target objects across model groups. As each target object corresponds to a subgoal, a larger number of target objects indicates higher task complexity. Overall, both completion rate and success rate decline as task complexity increases.

Table 15: Results under different prior-information settings on shared workspace tasks. Full prior includes both object appearance and location information. Without appearance removes object appearance description, without location removes prior location information, and no prior removes both.

| Model | Full Prior | | | Without Appearance | | | Without Location | | | No Prior | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SR | CR | AUC | SR | CR | AUC | SR | CR | AUC | SR | CR | AUC |
| GPT-5-mini | 0.917 | 0.982 | 0.776 | 0.865 | 0.955 | 0.730 | 0.344 | 0.691 | 0.436 | 0.271 | 0.609 | 0.431 |
| Gemma4-31B | 0.833 | 0.945 | 0.688 | 0.833 | 0.937 | 0.662 | 0.229 | 0.589 | 0.347 | 0.302 | 0.570 | 0.356 |
| Qwen3-32B-VL | 0.802 | 0.934 | 0.694 | 0.802 | 0.923 | 0.655 | 0.260 | 0.591 | 0.365 | 0.344 | 0.688 | 0.456 |

Table 16: Team-size scaling without location priors in a shared workspace (broadcast). Best results within each model are bold.

| Model | #Agents | SR | CR | AUC |
|---|---|---|---|---|
| Qwen3-8B-VL | 1 | 0.031 | 0.162 | 0.103 |
| | 2 | 0.021 | 0.199 | 0.101 |
| | 3 | **0.094** | 0.298 | **0.168** |
| | 4 | 0.062 | **0.307** | 0.163 |
| Qwen3-32B-VL | 1 | 0.219 | 0.520 | 0.344 |
| | 2 | 0.260 | 0.591 | 0.365 |
| | 3 | **0.292** | **0.657** | **0.404** |
| | 4 | 0.219 | 0.608 | 0.374 |

Table 17: Shared workspace performance with noisy location priors.

| Model | #Agents | SR | CR | AUC |
|---|---|---|---|---|
| GPT-5-mini | 1 | 0.760 | 0.902 | 0.719 |
| | 2 | 0.823 | 0.950 | 0.709 |
| Qwen3.5-9B | 1 | 0.781 | 0.911 | 0.665 |
| | 2 | 0.812 | 0.903 | 0.663 |
| Qwen3-32B-VL | 1 | 0.542 | 0.750 | 0.526 |
| | 2 | 0.698 | 0.878 | 0.620 |

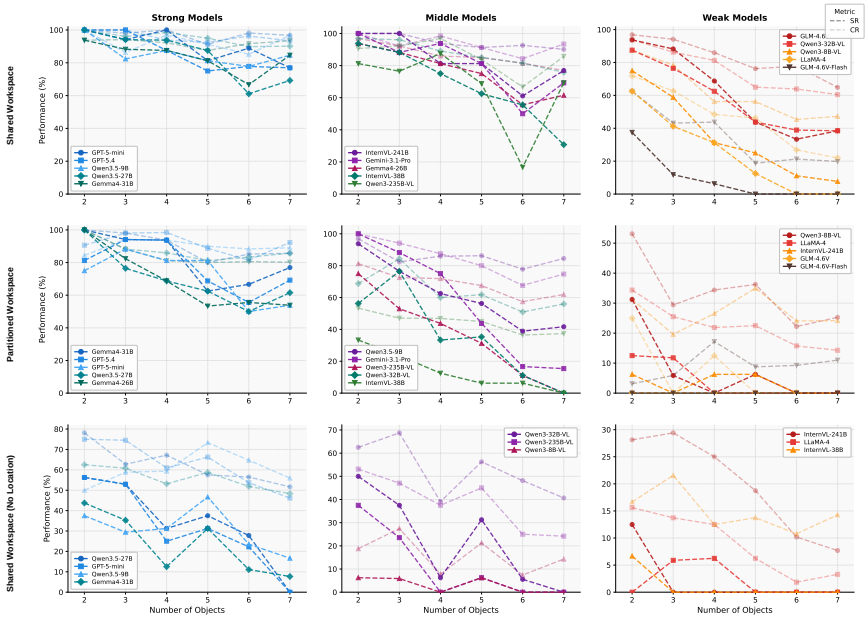



Figure 25: Performance by number of objects across task conditions and model groups.

# E EMERGENT COLLABORATIVE BEHAVIOR

By analyzing the communication and action trajectories under the basic broadcast protocol, we identify several emergent collaborative behavior patterns. In addition to the analysis in Section 4.3.2, this section provides representative examples and further details.

## E.1 EXAMPLES

Table 18 presents representative examples of collaborative behaviors, with the relevant text spans highlighted in yellow.

Table 18: Examples of collaborative behaviors emerge in the agent interactions in MECoBench under broadcast protocol.

| Category | Behavior | Example |
|---|---|---|
| *Information Sharing* | *Info-Plan* | "I will grab the two puddings and check the fridge for the wine. Could you go to the living room to find the cupcake?" |
| | *Info-Loc* | "I am in the bathroom and will head to the kitchen to look for the pudding and cupcake." |
| | *Info-Obj* | "I see the cupcake on the desk. I will grab it and then head to the living room." |
| *Task Coordination* | *Act-Del* | "Please go to the kitchen to grab the second apple, and then we can meet at the coffee table." |
| | *Task-Assign* | "I'll handle the kitchen and the bathroom." |
| *Alignment & Correction* | *Coord-Q.* | **State confirmation:** "Do you have the cupcake and the second apple? If not, please check the fridge." |
| | | **Location confirmation:** "I'm at door 287 holding the cupcake. Are you at door 287 and ready to receive? Please confirm." |
| | *Correct.* | **Self-correction:** "I accidentally grabbed a waterglass instead of the wineglass. I will put it back and then get the wineglass." |
| | | **Peer-correction:** "Agent 0, I see you're holding milk, but we need the juice. Please drop the milk and grab the correct juice from the kitchen table." |

## E.2 DETAILED ANALYSIS

**Coordination Queries Indicate Collaboration Challenges.** Episodes containing coordination queries have lower SR in both workspace settings (Table 19). To further investigate the underlying cause, we introduce two behavioral indicators of collaboration difficulty: negative report (*Neg.*), where an agent reports failing to locate a target object at a specific location, and *stuck*, where an agent reports being stuck in middle. As shown in Table 19, cases with coordination query involve more objects on average and exhibit substantially higher stuck and negative report rate. This pattern holds across both task types, suggesting that coordination query serves as a reactive signal to genuine collaboration obstacles.

Table 19: Coordination queries are associated with harder coordination states. *Q.* denotes coordination query.

| Workspace | Group | #Case | Avg. #Obj. | SR (%) | Stuck (%) | Neg. (%) |
|---|---|---|---|---|---|---|
| Shared Workspace | w/ Q. | 370 | 4.70 | 64.6 | 8.4 | 11.1 |
| | w/o Q. | 1727 | 4.35 | 74.7 | 2.0 | 4.5 |
| Partitioned Workspace | w/ Q. | 591 | 4.51 | 32.3 | 3.9 | 6.6 |
| | w/o Q. | 627 | 4.30 | 47.8 | 1.6 | 3.2 |

**Correction Helps Recover from Errors** Correction is associated with lower SR but higher CR in the overall comparison (Table 20), suggesting a potential role in error recovery. To examine this effect, we analyze its co-occurrence with wrong grabs (WGs), defined as actions that grasp task-irrelevant objects. Table 20 shows that episodes with correction contain substantially more WGs than those without, indicating that correction is mainly triggered by execution errors. Since WG episodes with correction also exhibit greater error severity, direct comparison may underestimate its recovery benefit. We therefore stratify WG episodes by error count and compute a weighted adjusted effect using shared stratum weights for both groups (Table 21). After adjustment, correction is positively associated with both SR and CR across task types, supporting its potential role in error recovery without establishing a causal effect. Stratum-level results further show larger gains under higher WG counts, suggesting that correction is especially useful when errors are more severe.

Table 20: Overall and wrong-grab-conditional effects of correction. *Corr.* and *WG* denote correction and wrong grab. *WG-Cond.* denotes cases where at least one wrong grab happened.

| Workspace | Cond. | Group | SR (%) | CR (%) | #WG |
|---|---|---|---|---|---|
| Shared Workspace | Overall | w/ Corr. | 67.3 | 90.1 | 3.05 |
| | | w/o Corr. | 74.1 | 88.5 | 1.37 |
| | | $\Delta$ | $-6.7$ | $+1.6$ | – |
| | WG | w/ Corr. | 61.1 | 88.3 | 3.79 |
| | | w/o Corr. | 66.2 | 87.0 | 2.84 |
| | | Raw $\Delta$ | $-5.1$ | $+1.3$ | – |
| | | Adj. $\Delta$ | $\mathbf{+2.4}$ | $\mathbf{+3.5}$ | – |
| Partitioned Workspace | Overall | w/ Corr. | 29.8 | 67.4 | 2.68 |
| | | w/o Corr. | 41.1 | 59.8 | 0.76 |
| | | $\Delta$ | $-11.3$ | $+7.7$ | – |
| | WG | w/ Corr. | 26.9 | 67.2 | 4.33 |
| | | w/o Corr. | 21.1 | 53.4 | 3.02 |
| | | Raw $\Delta$ | $+5.8$ | $+13.8$ | – |
| | | Adj. $\Delta$ | $\mathbf{+11.8}$ | $\mathbf{+15.4}$ | – |

Table 21: Correction effects in shared workspace tasks stratified by the number of wrong grabs.

| #WG | Group | #Case | SR (%) | CR (%) | $\Delta$SR/CR |
|---|---|---|---|---|---|
| 1 | Corr. | 70 | 82.9 | 95.2 | $+2.2/+3.7$ |
| | No Corr. | 351 | 80.6 | 91.5 | |
| 2 | Corr. | 53 | 69.8 | 89.3 | $-6.2/-1.5$ |
| | No Corr. | 200 | 76.0 | 90.8 | |
| 3 | Corr. | 44 | 68.2 | 91.9 | $+6.9/+4.1$ |
| | No Corr. | 80 | 61.3 | 87.8 | |
| 4+ | Corr. | 118 | 41.5 | 82.4 | $+7.4/+6.8$ |
| | No Corr. | 208 | 34.1 | 75.6 | |
| Adjusted effect | | | | $\Delta$SR/CR | $\mathbf{+2.4/+3.5}$ |

# F Failure Analysis

We illustrate the five failure mechanisms in Section 4.3.3 with trace-verified cases from two-agent broadcast teams (Table 22). Steps are zero-indexed; outcomes use the 30-round Shared Workspace and 40-round Partitioned Workspace budgets. All five cases fail (SR $= 0$); CR is reported as completed/required goal units.

## F.1 Examples

Table 22: Representative examples of five collaboration failure mechanisms. Quotes are trace excerpts; actions are abbreviated. S/P denote Shared/Partitioned Workspace.

| Failure Mode | Subtype | Representative Trace Excerpt |
|---|---|---|
| **Task Coordination** | Conflict actions / redundant work | **InternVL3.5-241B; S, case 5; CR** $= 2/6$**.**<br>**Step 3, A0 & A1:** Both execute `grab alcohol(196)`; both receive `Success`.<br>**Steps 6, 15–16:** A0 places this object on `coffeetable(318)`; A1 later grabs it and puts it back on the same table.<br>$\Rightarrow$ Same-object contention and repeated work; `alcohol` also fails to satisfy the `wine` goal. The simulator does not reject the simultaneous grabs. |

Table 22 (continued).

| Failure Mode | Subtype | Representative Trace Excerpt |
|---|---|---|
| **Task Coordination** | Commitment–execution mismatch | **InternVL3.5-38B; S, case 17; CR** $= 4/6$**.** **Step 13, A0:** *"I will grab the waterglass from the kitchentable and put it in the dishwasher."* **Raw proposal:** `grab cutleryfork`. **Parsed action:** `grab cutleryknife(139)` (`Success`). ⇒ The commitment already differs from the raw proposal, before the additional object-resolution error. The waterglass remains unfinished at the cutoff. |
| | Dependency coordination failure | **Gemini-3.1-Pro; P, case 38; CR** $= 2/4$**.** **Step 29, A0:** *"Please let me know when you arrive and are ready to receive."* A0 simultaneously attempts to hand over `cupcake(376)`. **Feedback:** `Characters are too far apart (4.5m), max 2.0m`. ⇒ Five of six handover attempts fail due to distance or a full receiving hand; the cupcake handover succeeds at step 31. |
| **Information Reliability** | State-inconsistent reporting | **GLM-4.6V; P, case 7; CR** $= 1/2$**.** **Step 8, A1:** *"Agent 0, I'm in the bedroom and have placed the wine on the coffee table."* **Trace check:** A1's earlier grab fails; its holding records remain empty through step 8, with no successful wine pickup or placement. ⇒ A1 reports a placement unsupported by its execution history. Only the cupcake goal is completed within budget. |
| | Message-linked error propagation | **GPT-5-mini; P, case 93; CR** $= 5/6$**.** **Step 20, A0:** *"Task is already complete"*, although only one of two required cellphones is on the desk. **A1 memory:** *"Agent 0 announced the task is complete and asked other agents to stop further moves."* **A1 action:** `wait`, explicitly justified by A0's report in its reasoning. At step 24, A1 rebroadcasts completion and requests cancellation of movement and handover. ⇒ A false report enters a teammate's memory, informs its action, and is repeated; the second cellphone remains missing. |

These mechanisms can co-occur; the traces do not establish exclusive causes or counterfactual causal effects. Table 4 reports rule-based detected case frequencies and associations with task success.

# G Sensitivity and Generalization Analysis

This section examines sensitivity to prior information, dialogue window size, and prompt design, followed by cross-simulator validation on TeamCraft Long et al. (2024).

## G.1 Dependence on Prior Information

**Which priors affect task performance?** We ablate object appearance descriptions and location information, comparing full priors, appearance-only priors, location-only priors, and no priors. Figure 26(a) show that removing locations causes the largest performance losses, showing that locating targets is a major bottleneck.

**Do changes in priors alter collaboration gains?** we further investigate the collaboration under two realistic settings: (1) *Missing location priors*: object locations are removed from the prior information, requiring agents to find targets through visual search. (2) *Noisy location priors*: an additional 30% false object locations are injected into the provided location.

Without location priors, two agents collaboration improve CR for every model and SR for most models in Figure 26(b), showing collaboration can support task progress under exploration demands. Table 23 demonstrates that collaboration achieves the largest relative performance gains when the information contains noise, suggesting that agents can compensate for misleading information

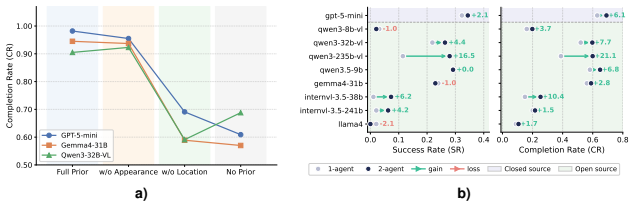



Figure 26: **Prior information and collaboration gains on shared workspace tasks.** (a) Completion rate under different prior-information settings. (b) Single-agent and two-agent performance without prior location information; arrows indicate absolute changes in percentage points.

Table 23: **Performance under noisy and incomplete task information.** Location priors are noisy (W/ Noise), clean (W/o Noise), or absent (W/o Loc.). Cells show relative SR gain above 1-agent SR → 2-agent SR.

| Model | W/ Noise | W/o Noise | W/o Loc. |
|---|---|---|---|
| GPT-5-mini | **+8.22%**<br>76.04→82.29 | **+1.16%**<br>90.62→91.67 | **+6.47%**<br>32.29→34.38 |
| Qwen3-32B | **+28.83%**<br>54.17→69.79 | **+28.57%**<br>58.33→75.00 | **+18.88%**<br>21.88→26.04 |
| Qwen3.5-9B | **+4.01%**<br>78.12→81.25 | **+3.52%**<br>85.42→88.54 | **+0.00%**<br>29.17→29.17 |

through communication and distributed exploration. These results together validate that **multi-agent collaboration is robust to insufficient and noisy information**.

### G.2 Sensitivity to Dialogue Window Size

We randomly sample 30 shared workspace task cases and evaluate Qwen3-32B-VL with different dialogue window sizes. As shown in Table 24, performance varies moderately across window sizes, with no consistent monotonic trend. The two-agent team outperforms the single-agent baseline in SR, CR, and AUC under every tested window size, indicating that **the collaboration benefit is robust to dialogue context length within the tested range**.

Table 24: **Sensitivity to dialogue window size.** Results for Qwen3-32B-VL on 30 randomly sampled shared workspace task cases. SR denotes success rate, CR denotes completion rate, and AUC denotes Step-CR AUC. All three metrics are reported as percentages. The single-agent baseline has no inter-agent dialogue window.

| #Agents | Dialogue Window Size | SR | CR | AUC |
|---|---|---|---|---|
| 1 | – | 60.00% | 73.01% | 52.32% |
| 2 | 5 | 80.00% | 91.43% | 69.39% |
| 2 | 10 | 86.67% | 95.32% | 71.28% |
| 2 | 20 | 80.00% | 94.40% | 67.58% |
| 2 | 30 | 86.67% | 94.72% | 67.53% |

### G.3 Sensitivity to Prompt Design

We additionally use GPT-5.6-sol to rewrite the system and user prompts for both the communication and action phases while preserving their core semantics. Table 25 compares the original and rewritten prompts. The changes in SR correspond to approximately one or two successful cases: rewriting increases shared workspace task SR by 3.33 percentage points for both single-agent and two-agent execution, while partitioned workspace task SR decreases by 6.67 points. Nevertheless, **the collaboration benefit on shared workspace tasks persists under both prompt variants**, with two-agent SR exceeding the single-agent baseline by 26.67 percentage points in each comparison.

### G.4 Cross-Simulator Validation

On TeamCraft (Long et al., 2024) smelting tasks, two-agent broadcast achieves higher CR and SR than single-agent and no-communication execution for both tested Qwen3-VL models under matched total step budgets.

To examine whether collaboration and communication gains extend beyond VirtualHome, we evaluate Qwen3-VL-8B and Qwen3-VL-32B on 20 TeamCraft (Long et al., 2024) smelting tasks in Minecraft. We include only tasks solvable by Bot 1 alone and compare single-agent execution, two-agent broadcast, and two-agent execution without communication under equal total step budgets: 20 steps for one agent and 10 steps per agent for two-agent teams.

Table 25: **Sensitivity to prompt design.** Results under original and rewritten system and user prompts. SR, CR, and Step-CR AUC are reported as percentages.

| Workspace | #Agents | Prompt | SR | CR | AUC |
|---|---|---|---|---|---|
| Shared Workspace | 1 | Rewritten | 63.33% | 80.20% | 62.97% |
| | 1 | Original | 60.00% | 73.01% | 52.32% |
| | 2 | Rewritten | 90.00% | 97.30% | 70.35% |
| | 2 | Original | 86.67% | 95.32% | 71.28% |
| Partitioned Workspace | 2 | Rewritten | 33.33% | 58.32% | 38.57% |
| | 2 | Original | 40.00% | 63.60% | 38.17% |

Table 26: **Results on TeamCraft smelting tasks.** Completion rate (CR) and success rate (SR) are reported as fractions. Average total steps sum the steps taken by all agents. Bold denotes the best CR and SR within each model.

| Model | Mode | Episodes | CR | SR | Avg. Total Steps |
|---|---|---|---|---|---|
| Qwen3-VL-8B | Single Agent | 20 | 0.2281 | 0.1000 | 18.53 |
| | 2-Agent Broadcast | 20 | **0.4825** | **0.4000** | 16.94 |
| | 2-Agent No-Comm | 20 | 0.4211 | 0.2500 | 16.74 |
| Qwen3-VL-32B | Single Agent | 20 | 0.8000 | 0.6500 | 11.20 |
| | 2-Agent Broadcast | 20 | **0.8917** | **0.8000** | 12.50 |
| | 2-Agent No-Comm | 20 | 0.8083 | 0.7000 | 13.70 |

Table 26 shows that broadcast teams achieve the highest CR and SR for both models. Broadcast raises SR from 10% to 40% for Qwen3-VL-8B and from 65% to 80% for Qwen3-VL-32B. It also outperforms two-agent execution without communication by 15 and 10 percentage points in SR, respectively. These results provide additional evidence for the benefits of collaboration and communication in a second environment, within the scope of the evaluated smelting tasks.