SYNAPSE-BENCH: STRUCTURED MULTI-AGENT COORDINATION UNDER MULTIMODAL PARTIAL OBSERVABILITY

Anonymous authors
Paper under double-blind review

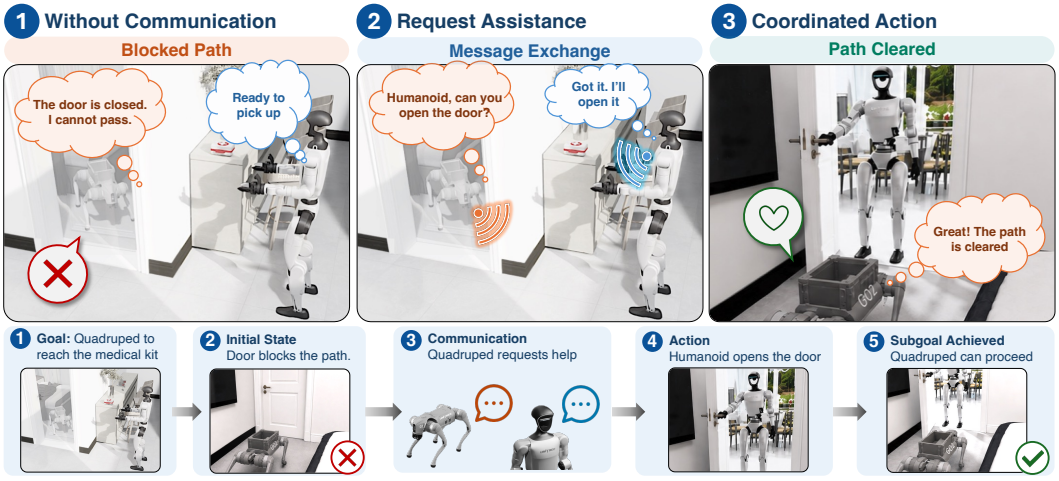


Figure 1: **When perception fails, communication resolves it.** The quadruped is blocked by a closed door out of the humanoid's sight, preventing progress. Through a single message, the humanoid updates its belief, opens the door, and clears the path. This example illustrates how communication resolves partial observability and enables cooperative task completion.

ABSTRACT

Embodied agents are increasingly expected to operate in multi-agent environments, where effective collaboration requires not only perception and control, but also communication and shared reasoning under partial observability. However, *how state representation affects heterogeneous collaboration under distributed observations remains insufficiently understood.* In this work, we introduce **SYNAPSE-Bench**, a new benchmark for evaluating *high-level decentralized coordination* in heterogeneous multi-agent systems. **SYNAPSE-Bench** features teams of agents with different embodiments, including humanoid and quadruped robots, operating in indoor environments with both egocentric and exocentric visual inputs. Agents interact over multiple timesteps, generating both *high-level actions from embodiment-conditioned action spaces* and communication messages to accomplish shared tasks under decentralized execution. A key design principle of **SYNAPSE-Bench** is using structured state representations to study *coordination under partial observability*. We incorporate scene graphs as an explicit intermediate representation of objects and their relation, and formulate an evaluation setting that compares unstructured histories, local scene graphs, and fused scene graphs as decision interfaces. To support systematic study, we provide GRAPHSTITCH as a reference approach for aligning and fusing scene graphs across agents. We conduct extensive zero-shot evaluations and controlled ablations over communication and representation design. Our experimental results show that local structured representations improve task success for several VLM backbones, while graph fusion yields model-dependent gains and regressions. **SYNAPSE-Bench** provides a common evaluation framework for studying how state representation, communication, and model assignment shape embodied multi-agent coordination.

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.

1 INTRODUCTION

Embodied artificial intelligence (Embodied AI) (Feng et al., 2026; Pfeifer & Iida, 2004; Duan et al., 2022; Liu et al., 2025b) has witnessed rapid progress over the past decades, driven by advances in perception, control, and learning-based decision-making (Liu et al., 2023; Li et al., 2023; Gao et al., 2022; Shridhar et al., 2020). Modern embodied agents are increasingly expected to operate in open-world environments, where they must perceive, reason, and act under uncertainty. In parallel, the emergence of generalist models, particularly large language models (LLMs), vision-language models (VLMs), and vision-language-action (VLA) models, has significantly expanded the scope of embodied intelligence (Achiam et al., 2023; Team, 2025; Bjorck et al., 2025; Collaboration et al., 2023; Driess et al., 2023; Brohan et al., 2023). These models exhibit strong capabilities in multimodal understanding, commonsense reasoning, and instruction-following, making them promising candidates to serve as the decision-making "brain" of embodied systems.

Despite these advances, evaluating embodied intelligence in real-world settings remains a fundamental challenge. Existing benchmarks primarily focus on individual capabilities such as navigation, manipulation, or perception (Yang et al., 2025; Zheng et al., 2022; Chen et al., 2025b; Xu et al., 2025; Majumdar et al., 2024; Li et al., 2024), and are often designed for *single-agent* settings. However, real-world deployment increasingly involves *multi-agent systems*, where multiple robots with different embodiments must coordinate to achieve shared goals. Recent progress in robotic platforms, including humanoid and quadruped systems (Bjorck et al., 2025; Sheng et al., 2025; Lykov et al., 2024), further highlights the need for evaluation frameworks that capture *heterogeneous collaboration* under partial observability.

This setting motivates evaluating how agents use distributed observations to coordinate under asymmetric capabilities. Each agent has limited, viewpoint-dependent observations, requiring coordination through communication and shared reasoning (cf. Figure 1). At the same time, differences in embodiment and sensing capabilities create asymmetries in perception and action, making it nontrivial to maintain a consistent understanding of the environment across agents. As a result, effective collaboration requires not only strong perception and control but also *shared state construction* and *multi-agent reasoning* under uncertainty.

Existing approaches fail to capture this joint problem. Recent work on multi-agent collaboration with LLMs (Zhang et al., 2024; Liu et al., 2025a) shows that language enables coordination, but often lacks grounding in visual observations. Conversely, reinforcement learning methods incorporate visual inputs in simulated environments (Liu et al., 2023; Duan et al., 2022), yet typically focus on single-agent or homogeneous settings and rely on implicit representations without explicitly modeling shared world structure. As summarized in Table 1, prior work can be broadly categorized into (i) LLM/VLM-based embodied agents, (ii) heterogeneous multi-robot systems, and (iii) embodied benchmarks for perception or structured reasoning. While each line of work captures important aspects of embodied intelligence, none jointly address heterogeneous embodiment, decentralized multi-agent coordination, multimodal perception, and explicit structured state representation within a unified evaluation framework.

To bridge this gap, we present a new benchmark, Structured sYstems for multi-ageNt collAboration with Perception and communication in Embodied settings (**SYNAPSE-Bench**), for evaluating collaborative embodied intelligence in heterogeneous multi-agent settings. The benchmark features a team of agents, including a humanoid and a quadruped, operating in indoor environments with both egocentric and exocentric visual observations under partial observability. Agents must generate both actions and communication messages over multi-turn interactions to accomplish shared tasks. This setup explicitly evaluates the integration of perception, reasoning, and communication within a decentralized multi-agent framework.

A central design principle of **SYNAPSE-Bench** is that *structured state representations are essential for evaluating coordination under communication constraints* (Gu et al., 2025; Marcotte et al., 2020; Bayer & Faigl, 2026). In realistic multi-agent systems, agents cannot freely exchange raw sensory inputs due to bandwidth and latency limitations (Bayer & Faigl, 2026; Marcotte et al., 2020). Transmitting full RGB observations is expensive, while aggressively compressing such data often discards high-level semantics critical for decision-making (Xue et al., 2026). As a result, effective collaboration requires representations that are both *compact* and *semantically expressive*.

Table 1: Comparison of representative embodied AI methods and benchmarks. We group prior work by its primary focus: *LLM/VLM-based embodied agents*, *heterogeneous multi-robot systems*, and *embodied benchmarks for perception or structured reasoning*. A checkmark (✓) indicates that the corresponding property is explicitly instantiated in the task setting or evaluation protocol, rather than merely being compatible with the method. **SYNAPSE-Bench** jointly covers multi-view perception, capability-asymmetric humanoid–quadruped teaming, decentralized multi-agent decision-making, collaborative dependency, and explicit scene-graph state representation. **Modal.** denotes input modalities explicitly provided to the agent, including Tr text, 🖼 images/videos, and accompanied 🕸 scene graphs. * denotes the number used during evaluation.

Work	Modal.	Num. Tasks	Viewpoints Global	Viewpoints Local	Agent Type Humanoid	Agent Type Quadruped	Multi Agent	Decen-tralized	Hetero-geneous	Colla-borative	Scene Graph
LLM/VLM-based embodied agents / reasoning-centric methods											
TwoStep (Bai et al., 2024)	Tr	20	✓	✗	✗	✗	✓	✗	✗	✓	✗
ZeroCAP (Venkatesh & Min, 2025)	Tr	15	✓	✗	✗	✗	✓	✗	✗	✓	✗
CoELA (Zhang et al., 2024)	🖼, Tr	34	✗	✓	✗	✗	✓	✓	✗	✓	✗
MacNet (Qian et al., 2025)	Tr	–	✓	✗	✗	✗	✓	✗	✗	✓	✗
REMAC (Yuan et al., 2025)	🖼, Tr	40*	✗	✓	✗	✗	✓	✓	✗	✓	✗
Heterogeneous multi-robot systems / system-centric benchmarks											
RoCoBench (Mandi et al., 2024)	🖼, Tr	6	✓	✗	✗	✗	✓	✗	✗	✓	✗
COHERENT (Liu et al., 2025a)	🖼, Tr	100	✓	✗	✗	✓	✓	✗	✓	✓	✗
RoboOS (Tan et al., 2025b)	🖼, Tr	68	✓	✓	✓	✓	✓	✗	✓	✓	✓
RoboFactory (Qin et al., 2025)	🖼, Tr	11	✓	✗	✗	✗	✓	✗	✗	✓	✗
CLiMBench (Song et al., 2025)	🖼, Tr	4*	✓	✗	✓	✗	✓	✗	✓	✓	✗
MHRC (Yu et al., 2024)	Tr, 🕸	12*	✓	✗	✗	✗	✓	✓	✓	✓	✓
Embodied VLM benchmarks / perception and structured reasoning											
ENACT (Wang et al., 2025a)	🖼, Tr, 🕸	29	✗	✓	✗	✗	✗	✗	✗	✗	✓
ESCA (Huang et al., 2026)	🖼, Tr, 🕸	–	✗	✓	✗	✗	✗	✗	✗	✗	✓
SYNAPSE-Bench (ours)	🖼, Tr, 🕸	**461**	✓	✓	✓	✓	✓	✓	✓	✓	✓

To capture this trade-off, **SYNAPSE-Bench** introduces *scene graphs* as an explicit intermediate representation for communication and reasoning. Each agent extracts a structured graph from its local observation, encoding objects, attributes, and relations in a form that is significantly more bandwidth-efficient than raw visual data while preserving key semantic information. Compared to unstructured visual-text histories, scene graphs provide a more interpretable and compositional abstraction of the environment, enabling agents to communicate partial knowledge that is both concise and actionable (Johnson et al., 2015; Zellers et al., 2018; Yang et al., 2018; Ju et al., 2026).

Crucially, **SYNAPSE-Bench** treats structured representations not merely as an input modality, but as a *first-class evaluation axis*. The benchmark is designed to study how agents construct, exchange, and utilize *partial scene graphs* under decentralized execution. Since each agent observes only a fragment of the environment, the resulting graphs are inherently incomplete and inconsistent across viewpoints. This introduces a fundamental challenge: how to reconcile partial and distributed structured observations into a coherent shared understanding that supports coordinated action.

To facilitate systematic evaluation, we provide a reference fusion module, GRAPHSTITCH, which aligns and integrates partial scene graphs across agents. While GRAPHSTITCH serves as a *concrete instantiation*, the benchmark is *agnostic to the choice of fusion strategy*, enabling future work to explore alternative approaches to structured communication. Through this design, **SYNAPSE-Bench** enables the study of a previously underexplored dimension of embodied intelligence: *communication-efficient, structure-aware multi-agent reasoning under partial observability*.

Contributions of this Work. This work has five main contributions. First, a new benchmark **SYNAPSE-Bench** is presented for evaluating collaborative embodied intelligence in heterogeneous multi-agent settings, featuring multimodal perception, partial observability, and multi-turn communication between agents. Second, we identify *communication-efficient structured state representation* as a critical yet underexplored dimension of embodied AI, and incorporate scene graphs as a first-class modality that captures the trade-off between bandwidth efficiency and semantic expressiveness. Third, a new evaluation setting is formalized for *structure-aware multi-agent reasoning*, where agents must construct, exchange, and utilize partial structured observations under decentralized execution. Fourth, we provide GRAPHSTITCH as a reference method for aligning and fusing partial scene graphs across agents, enabling systematic evaluation while remaining model-agnostic. Finally, we conduct zero-shot evaluations and targeted ablations on communication and representation design, using GRAPHSTITCH to analyze multi-agent state integration and show that structured representations substantially enhance multi-agent coordination and downstream task performance.

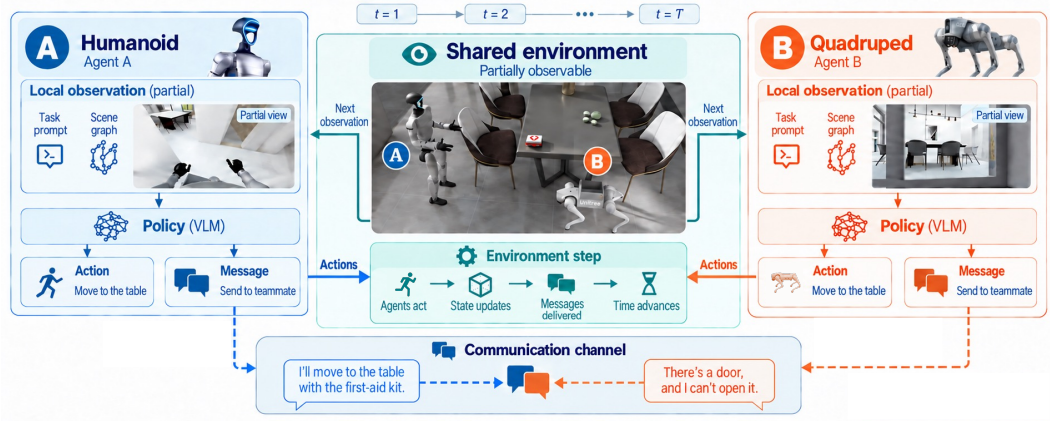


Figure 2: **SYNAPSE-Bench Overview:** Multi-agent embodied coordination under partial observability. Two heterogeneous agents operate with partial observations and iteratively produce actions and natural-language messages over time. Their interactions are mediated through a shared environment and a communication channel, forming a closed decision loop across timesteps.

2 SYNAPSE-BENCH – THE OVERVIEW

We introduce **SYNAPSE-Bench**, a benchmark for evaluating multi-agent embodied coordination under partial observability (overview in Figure 2). Our design centers on three principles: (i) *role-differentiated coordination*, where observations and assigned actions are distributed across agents; (ii) *heterogeneous embodiment*, which induces asymmetric capabilities and roles; and (iii) *multi-turn interaction*, requiring coordinated reasoning and communication over time.

2.1 PROBLEM SETTING

Decentralized multi-agent interaction. We consider a decentralized partially observable Markov decision process (Dec-POMDP) (Bernstein et al., 2002) defined as $\langle \mathcal{S}, \{\mathcal{A}^i\}_{i=1}^N, \{\mathcal{O}^i\}_{i=1}^N, T, R \rangle$, where N agents interact in a shared environment with state $s_t \in \mathcal{S}$. Each agent i receives a local observation $o_t^i \in \mathcal{O}^i$ and selects an action $a_t^i \in \mathcal{A}^i$. The environment evolves according to $T(s_{t+1} \mid s_t, \mathbf{a}_t)$ with joint action $\mathbf{a}_t = \{a_t^j\}_{j=1}^N$, and returns a task-level reward R. We focus on *heterogeneous* agents, where action spaces $\mathcal{A}^i$ and observation modalities $\mathcal{O}^i$ may differ across agents.

Multimodal observations and communication. Each observation o_t^i is multimodal, consisting of egocentric visual input (e.g., RGB frames) and textual context (e.g., task description and history). In addition to acting in the environment, agents communicate through a shared channel. At each timestep t, agent i produces a pair of outputs $(a_t^i, m_t^i) \sim \pi_i(\cdot \mid \tau_t^i)$, where a_t^i is a high-level action and m_t^i is a natural language message. The policy π_i is conditioned on the local interaction history $\tau_t^i = \{o_{1:t}^i, a_{1:t-1}^i, m_{1:t-1}^{\neg i}\}$, which includes both self-taken actions $a_{1:t-1}^i$ and messages received from other agents $m_{1:t-1}^{\neg i}$ from timestep 1 to $t-1$. This formulation captures decentralized decision-making with explicit communication under partial observability.

2.2 TASK DESIGN

We construct indoor collaborative tasks that combine navigation, object manipulation, and fetch-and-deliver skills. Although these skills are individually well studied (Li et al., 2023; Anderson et al., 2018; Shridhar et al., 2020), our episodes couple them through ordered, embodiment-specific roles under distributed egocentric observations. Each episode follows a manipulation–transport relay: the humanoid loads the target object and the quadruped transports it to the destination. The temporal position of humanoid door operations yields four mutually exclusive coordination structures: direct relay (291 episodes), access before loading (41), egress after loading (112), and access both before and after loading (17). Together, these families cover all 461 episodes (Figure 5A).

Inter-agent dependency arises through three complementary mechanisms: (1) **ordered dependencies**, in which loading or door clearance enables a later action by the other agent; (2) **embodiment asymmetry**, which assigns object and door manipulation to the humanoid and transport to the quadruped (Figure 5B); and (3) **observation asymmetry**, under which spatially separated agents receive complementary local evidence. Consequently, coordination may require communicating target

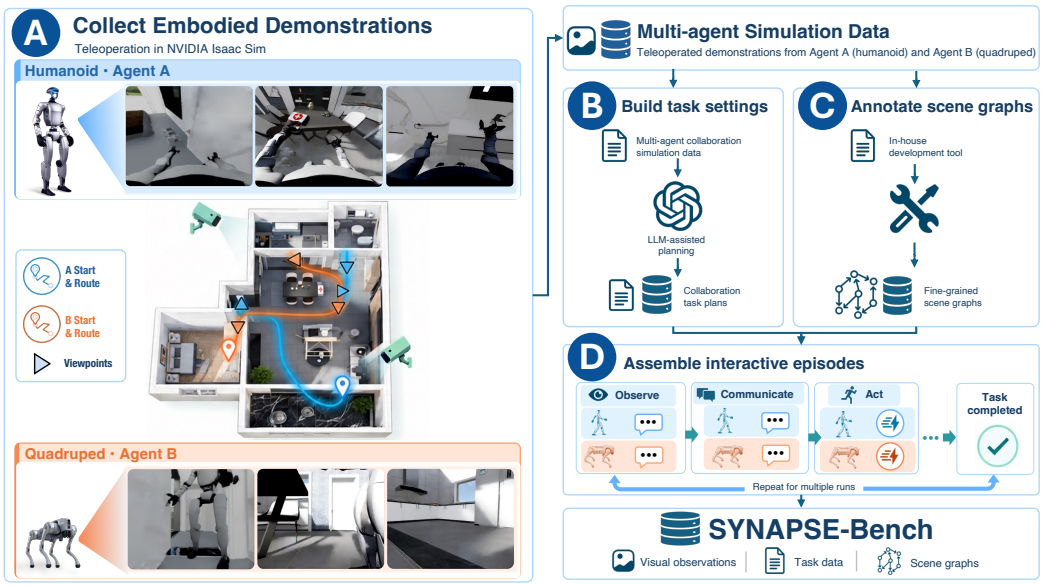


Figure 3: **Overview of the data collection pipeline**, including (A) trajectory-level observation capture, (B) annotation of executable actions and communication, (C) scene graph construction, and (D) episodic task generation.

identity and location, obstacle state, readiness, and subgoal completion. Episodes remain compact (3–10 paired decision steps), but door-conditioned families introduce additional enabling actions and agent alternation rather than merely longer navigation. Please see Appendix A for more details.

2.3 Environment and Agents

Agents. The benchmark specifies a heterogeneous team comprising a *humanoid* and a *quadruped*. Within the benchmark protocol, the humanoid is assigned articulated manipulation actions, including object and door interaction, while the quadruped is assigned navigation and transport actions using its mounted carriage. This asymmetry induces complementary roles within the benchmark protocol and lets us evaluate capability-aware action selection and coordination. The evaluated interface abstracts away motion planning and fine-grained control.

Observations. Each agent operates under partial observability with egocentric visual input, consisting of RGB frames, task context, and interaction history. Due to occlusions, limited fields of view, and spatial separation, no agent has access to the full environment state, creating perceptual uncertainty and information asymmetry that require temporal reasoning and communication.

Interaction Protocol. Agents interact with the environment through a multi-turn rollout. At each timestep t, each agent receives its current observation and produces a pair of outputs: an action and a message. These outputs are executed simultaneously, and messages are shared across agents before the next timestep. Formally, each step follows the loop: $(o_t^i, \tau_t^i) \rightarrow (a_t^i, m_t^i) \rightarrow$ update environment $\rightarrow o_{t+1}^i$. This protocol captures decentralized decision-making with communication, where coordination emerges via repeated interaction rather than centralized planning.

2.4 Dataset Construction

Environment. We collect all data in NVIDIA Isaac Sim with PhysX-based physics simulation (NVIDIA, 2025) (cf. Figure 3). Our benchmark uses five indoor environments adapted from (SpatialVerse Research Team, 2025), covering diverse room layouts, object arrangements, and navigation constraints. Across episodes, we vary object placements and initial agent states. Objects are placed on reachable surfaces, while agents are initialized in plausible intermediate states, e.g., performing household activities, to create realistic starting conditions rather than artificial reset states.

Trajectory generation. Trajectories are collected from human operators with access to the task specification. Each task is decomposed into an ordered sequence of subgoals, where completion of subgoal t_k is required before subgoal t_{k+1} can proceed. Operators execute near-optimal behavior at the subtask level, ensuring that trajectories demonstrate successful coordination while preserving variation in interaction patterns.



Figure 4: **(A)** Structured agent record; **(B)** Example multi-agent episode roll-out. Note that the action and communication message are slightly simplified for brevity and clarity.

Annotation and verification. For each trajectory, we annotate both executable actions and communication messages. Actions are represented as textual commands designed to be directly interpretable by embodied agents, while communication is expressed in natural language with controlled communicative intent. Initial action-message annotations are generated with GPT-4o (Achiam et al., 2023) and subsequently verified by human experts. To ensure consistency, each action is checked against the corresponding simulator transition, and each message is verified to align with the agent's observation, intent, and executed behavior. The dataset is stored in JSON format as timestep-level samples, and a simplified sample is shown in Figure 4. For scene graph construction, we leverage 3D bounding box annotations from NVIDIA Isaac Sim and develop an in-house annotation tool to derive spatial relationships between objects. More details are discussed in Appendix A.

Quality control. We retain only episodes that are successful, temporally coherent, and semantically meaningful. Human experts manually filter trajectories to remove inconsistent action sequences, misleading communication, or interactions that do not reflect genuine cross-agent coordination.

2.5 Dataset Statistics

The benchmark contains 461 episodes and 3,746 agent-timestep samples collected across five indoor environments. Tasks span three categories: navigation, manipulation, and fetch-and-deliver. Episodes are designed to be compact (3–10 steps), requiring explicit inter-agent coordination. To support multimodal evaluation, the dataset provides 74,520 RGB frames across four viewpoints, along with 37,260 local scene graphs and 188,478 relation triplets derived from egocentric observations. These annotations enable systematic evaluation of structured state representations for multi-agent collaboration under partial observability.

3 Structured State Representation & Multi-Agent Graph Fusion

3.1 Motivation and Scene Graph Representation

In multi-agent embodied settings, each agent operates under partial observability and maintains its own local interaction history. Raw multimodal histories, which consists of visual observations, actions, and communication, are often unstructured and difficult to aggregate into a coherent belief. This limits coordination, as agents must reason not only about the environment but also about each other's latent knowledge.

To address this, we study whether *explicit structured state representations* can improve coordination by providing a compact, interpretable abstraction of the environment, and enabling more efficient information exchange across agents. We represent each agent's local observation as a *scene graph*: $G_t^i = (\mathcal{V}_t^i, \mathcal{E}_t^i)$, where $\mathcal{V}_t^i$ is a set of object nodes and $\mathcal{E}_t^i$ is a set of relations. Each node $v \in \mathcal{V}_t^i$ is associated with attributes such as object category. Each edge $(u, v) \in \mathcal{E}_t^i$ encodes semantic or

spatial relations. This structured abstraction enables explicit reasoning over object relationships and environmental layout, in contrast to raw perceptual histories.

3.2 Reference Multi-Agent Graph Fusion

While scene graphs provide structured local representations, each agent only observes a partial view of the environment. Consequently, local graphs are incomplete and may be inconsistent due to viewpoint differences, occlusions, and perception noise. The goal of graph fusion is to combine a set of local graphs $\{G_t^i\}_{i=1}^N$ into a shared representation G_t^* that better approximates the underlying environment state.

GraphStitch. We introduce GraphStitch as a lightweight reference method for fusing local scene graphs. The approach proceeds in three stages: (1) *node alignment*, establishing correspondences across graphs; (2) *constraint construction*, converting triplets into directional and distance-based constraints; and (3) *frame normalization and merging*, aggregating constraints in a shared frame to estimate a unified layout. The resulting fused graph, encoding a shared belief state, is serialized and provided to all agents. The detailed algorithm is stated in Appendix D.

Role in the Benchmark. Within SYNAPSE-Bench, GraphStitch serves as a reference baseline for structured information fusion. It is used to instantiate the fused-representation setting and to evaluate how aggregating partial observations impacts coordination performance. We emphasize that GraphStitch is *not optimized for task-specific performance*, but rather *provides a simple and reproducible mechanism for studying graph-based communication*.

4 Evaluation Protocol

We evaluate *multi-agent coordination under decentralized execution* with explicit communication, which follows a synchronized multi-agent roll-out, i.e., agents act and exchange at each timestep.

4.1 Metrics

Reference-Trajectory Pass Rate (RTPR). We measure the fraction of evaluated episodes in which the joint task is successfully completed within a fixed horizon. At each paired decision step, the roll-out advances when both agents jointly satisfy the environment-defined criteria. An episode passes if every paired step advances under this condition.

Normalized Decision-Attempt Agreement (NDA). We also use annotated paired decisions as a proxy for progress along the reference trajectory. For each logged decision attempt, a judge scores both agents' actions and communications against their reference annotations.

$$\mathrm{RTPR} = \frac{1}{|\mathcal{D}|}\sum_{\tau\in\mathcal{D}} \mathrm{Pass}_\lambda(\tau), \qquad \mathrm{NDA} = \frac{1}{|\mathcal{R}|}\sum_{\rho\in\mathcal{R}} \frac{1}{|\mathcal{A}_\rho|}\sum_{r\in\mathcal{A}_\rho} \mathrm{Pass}_\varepsilon(r), \tag{1}$$

where $\mathcal{D}$ denotes the evaluation set, $\mathcal{R}$ the evaluated rollouts, $\mathcal{A}_\rho$ the logged decision attempts in rollout ρ, and λ, ε the passing thresholds ($\lambda = 0.3, \varepsilon = 0.75$).

4.2 Experimental Settings

Model Variants. We evaluate a diverse set of VLMs, spanning both proprietary and open-source systems. Proprietary models include GPT-5.4 (OpenAI, 2026) and Gemini-2.5-Pro (Comanici et al., 2025). Our open-source suite comprises Qwen3.6-35B-A3B (Qwen Team, 2026b), InternVL-3.5-30B-A3B (Wang et al., 2025b), Gemma4-E4B-it (Google DeepMind, 2026), Eagle2.5-8B (Chen et al., 2026), MiniCPM-V 4.5 (Yu et al., 2025), GLM-4.6V-Flash (Hong et al., 2025), and Qwen3.5-9B (Qwen Team, 2026a). These models cover a broad range of scales, architectures, and training regimes, enabling a systematic evaluation of coordination performance under heterogeneous model capabilities and design choices (details in Appendix B).

Representation Settings. To evaluate the impact of structured representations, we consider three input conditions: ① *Unstructured*, where agents rely solely on raw visual-text observations; ② *Local Structured*, where each agent receives its own partial scene graph; and ③ *Fused Structured*, where a shared scene graph is constructed from multiple agents' observations. These settings enable controlled comparison of representation fidelity and information sharing.

Table 2: Comparison of model performance across three representation settings: *Unstructured* (`Ustrd`), *Local Structured* (`Lstrd`), and *Fused Structured* via GRAPHSTITCH (`Fstrd`).

Model	Ustrd		Lstrd		Fstrd	
	RTPR (%) ↑	NDA (%) ↑	RTPR (%) ↑	NDA (%) ↑	RTPR (%) ↑	NDA (%) ↑
Proprietary Models						
GPT-5.4 (OpenAI, 2026)	38.2	39.10	40.2	21.68	41.4	26.04
Gemini-2.5-Pro (Comanici et al., 2025)	26.7	19.14	27.4	14.92	24.6	12.85
Open-sourced Models						
Qwen3.6 (Qwen Team, 2026b)	28.2	24.94	38.2	25.95	41.8	27.87
InternVL3.5 (Wang et al., 2025b)	20.6	23.31	21.0	18.61	17.4	15.11
Gemma4-E4B-it (Google DeepMind, 2026)	31.8	37.60	33.8	27.24	33.8	30.02
Eagle2.5 (Chen et al., 2026)	31.0	15.68	50.4	43.98	47.8	53.14
MiniCPM-V 4.5 (Yu et al., 2025)	9.8	6.76	26.6	14.10	18.2	12.09
GLM-4.6V-Flash (Hong et al., 2025)	26.8	24.56	27.6	25.95	30.2	27.15
Qwen3.5 (Qwen Team, 2026a)	22.8	15.49	28.8	19.95	34.6	24.46

Communication Settings. We study the role of communication under two conditions: ① *Free-form Communication*, where agents generate messages at each timestep based on their local observations and interaction history (including past actions and exchanged messages); and ② *Oracle Communication*, where agents are provided with privileged, ground-truth signals to approximate an upper bound on performance under idealized information access.

Homogeneous vs. Heterogeneous VLMs Settings. We consider two multi-agent configurations: ① *Homogeneous*, where all agents share the same VLM backbone, and ② *Heterogeneous*, where agents are instantiated with different VLMs. This setting evaluates whether coordination strategies generalize across mismatched reasoning and perception capabilities.

5 EXPERIMENTAL RESULTS

5.1 MAIN RESULTS

Table 2 compares three representation settings across VLM backbones. Local scene graphs improve RTPR for every evaluated model, with gains ranging from 0.4 points for InternVL3.5 to 19.4 points for Eagle2.5. This consistent improvement suggests that explicitly representing objects and their relations provides a useful decision interface under partial observability. Fusion offers additional gains for several models: Qwen3.6's RTPR increases from 38.2% to 41.8%, while Qwen3.5's rises from 28.8% to 34.6%. Its effect nevertheless varies across backbones. For Eagle2.5, fusion raises NDA from 43.98% to 53.14%, while RTPR changes from 50.4% to 47.8%, illustrating that stronger agreement at individual steps need not produce a higher trajectory pass rate. Together, the results establish a consistent benefit from local structure and show that fused information can provide further gains, depending on how a model uses the shared information.

5.2 COMMUNICATION ABLATION

As shown in Table 3, free-form communication yields higher RTPR for all five backbones, while oracle messages yield higher NDA. This consistent difference shows that communication design affects coordination outcomes: providing more accurate information does not necessarily improve performance across a complete trajectory. A possible explanation is communication alignment: whether agents interpret messages consistently with their teammate's intended task state and next action. However, the ablation does not isolate alignment from differences in message format, timing, or consistency with agents' actions. These findings demonstrate the value of **SYNAPSE-Bench** for identifying coordination effects that a single performance metric would miss.

5.3 HOMOGENEOUS VS. HETEROGENEOUS VLMS

Table 4 shows that performance depends on VLM pairing and role assignment. Eagle2.5-8B paired with itself achieves the highest RTPR and NDA (50.4 / 43.98). Heterogeneous results vary: Gemma4 & Eagle2.5-8B remains competitive, while pairings with GLM-4.6V Flash score poorly. Thus, a strong shared backbone is a reliable choice among the tested configurations, while heterogeneous teams require careful model pairing and role assignment.

Table 3: Communication ablation: free-form (`fc`) vs. oracle (`oc`).

Backbone	Comm.	RTPR (%) ↑	NDA (%) ↑
Eagle2.5	`fc`	50.4	43.98
	`oc`	16.0	58.47
Gemma4-E4B-it	`fc`	33.8	27.24
	`oc`	19.6	58.94
MiniCPM-V 4.5	`fc`	26.6	14.10
	`oc`	9.2	43.21
GLM-4.6V-Flash	`fc`	27.6	25.95
	`oc`	11.4	40.67
InternVL-3.5	`fc`	21.0	18.61
	`oc`	12.4	43.41

Table 4: Homogeneous (△/△) vs. heterogeneous (△/□) VLM configurations.

Setting	Humanoid VLM	Quadruped VLM	RTPR (%) ↑	NDA (%) ↑
△/△	Eagle2.5	Eagle2.5	50.4	43.98
△/△	Gemma4	Gemma4	33.8	27.24
△/□	Eagle2.5	Gemma4	30.6	31.22
△/□	Eagle2.5	GLM-4.6V-Flash	1.2	3.07
△/□	Gemma4	Eagle2.5	40.0	31.70
△/□	Gemma4	GLM-4.6V-Flash	1.0	1.58

6 Related Work

Embodied AI Benchmarks. Embodied AI benchmarks have advanced navigation, manipulation, and household task completion under egocentric perception and partial observability (Li et al., 2023; Gao et al., 2022; Shridhar et al., 2020; Chang et al., 2025). However, many standard benchmarks focus on single-agent settings, leaving communication, heterogeneous embodiment, and inter-agent dependency only partially explored.

Multi-Agent Collaboration. MARL and robot-teaming methods study decentralized coordination, heterogeneous agents, and cooperative execution (Yu et al., 2023; Bettini et al., 2023; Gao et al., 2023). Recent LLM-based systems further address heterogeneous multi-robot collaboration, language-guided planning, and embodiment-aware coordination (Liu et al., 2025a; Tan et al., 2025b; Yu et al., 2024; Chen et al., 2025a). These works are closely related, but often rely on centralized planners, system-specific memory, or handcrafted abstractions rather than standardized evaluation of multimodal decentralized reasoning.

Vision-Language Models for Embodied Tasks. LLMs and VLMs enable embodied agents to decompose instructions, plan actions, and communicate through natural language (Bai et al., 2024; Zhang et al., 2024; Qian et al., 2025). These methods show strong potential for embodied coordination, but many still assume centralized control or simplified environments with limited embodiment diversity. Recent work (Zhang et al., 2025) more directly studies multi-agent cooperation under partial observability with compositional world models.

Structured Representations for Embodied Reasoning. Structured representations, e.g., scene graphs, shared memories, and world models, improve grounding by abstracting objects, relations, temporal events, and embodiment-specific capabilities (Huang et al., 2026; Yu et al., 2024; Tan et al., 2025b; Zhang et al., 2025; Tan et al., 2025a). However, their role in multi-agent coordination remains underexplored, especially when decentralized observations must be fused into an explicit shared state. Our work targets this gap by evaluating collaborative embodied agents under communication, partial observability, heterogeneous embodiment, and structured scene-level reasoning.

7 Conclusion

We introduced **SYNAPSE-Bench**, a benchmark for multi-agent coordination under partial observability with heterogeneous embodied agents. By jointly varying *communication* and *state representations*, it enables controlled study of their interaction in decentralized decision-making. Our results characterize how communication interface and state representation interact: communication enables one mechanism for exchanging otherwise distributed information, while structured representations (e.g., scene graphs and GraphStitch) determine how effectively that information is integrated. Evaluating them in isolation underestimates their combined impact, with the strongest gains when both are present. Despite these gains, performance remains far from saturated under realistic constraints, e.g., limited perception and ambiguity. Although limited by its simulated setting and short-horizon tasks, **SYNAPSE-Bench** isolates core coordination challenges in a controlled and reproducible manner, providing a foundation for jointly benchmarking communication and representation in multi-agent systems.

Discussion. Although constrained in scope, **SYNAPSE-Bench** captures fundamental coordination challenges: partial observability, inter-agent dependency, and communication under uncertainty. These properties persist beyond simulation, making the benchmark a useful testbed for studying core mechanisms of multi-agent reasoning before scaling to more complex and realistic settings.

AI USE STATEMENT

AI tools were used to assist with language editing, code implementation and debugging, and data generation. Data generation followed a human-in-the-loop process: human annotators reviewed and corrected the generated annotations before their inclusion in the benchmark. The authors also inspected the AI-assisted text, reviewed and tested the relevant code, and checked the reported results against the released data and evaluation outputs. Generative AI was used as an assistive tool rather than as an autonomous author or evaluator. The authors take full responsibility for the paper's content, claims, code, data, and artifacts.

ETHICS STATEMENT

SYNAPSE-Bench contains simulated indoor environments and robot interactions; it does not contain human-subject data or personally identifiable information. Accordingly, its release presents minimal direct privacy risk. The benchmark is intended to support research on transparent, communication-aware coordination for collaborative robots, with potential benefits in assistive robotics, logistics, and other settings in which agents must coordinate under partial observability. At the same time, advances in multi-agent perception and coordination may be repurposed for surveillance, military, or other high-risk applications. Results obtained in simulation may also overstate readiness for physical deployment because they do not capture the full range of real-world sensing errors, dynamics, failures, or human safety concerns. We therefore discourage using benchmark performance alone as evidence that a system is safe for deployment and recommend application-specific risk assessment, sim-to-real validation, human oversight, and appropriate physical safeguards. The dataset and code are released for non-commercial research under the accompanying license, and users are responsible for complying with that license and applicable laws.

REPRODUCIBILITY STATEMENT

We support reproducibility by releasing the evaluation split, the full benchmark data, scene assets, prompts, rollout-generation and scoring code, and the GRAPHSTITCH reference implementation. Section 2 describes the task and data construction, Section 4 specifies the evaluation settings and metrics, and Appendix A documents benchmark statistics, task families, action-label distributions, dependency structure, and scene-graph annotation. Appendix B reports the implementation and evaluation protocol, Appendix D gives the complete GRAPHSTITCH algorithm, and Appendix E provides the system prompts and output schemas.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

David Bai, Ishika Singh, David Traum, and Jesse Thomason. Twostep: Multi-agent task planning using classical planners and large language models. *arXiv preprint arXiv:2403.17246*, 2024.

Jan Bayer and Jan Faigl. Decentralized multi-robot exploration under low-bandwidth communications. *Autonomous Robots*, 50(1):7, 2026.

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, 2002.

Matteo Bettini, Ajay Shankar, and Amanda Prorok. Heterogeneous multi-robot reinforcement learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pp. 1485–1494, 2023.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, D Driess, A Dubey, C Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. arxiv. *arXiv preprint arXiv:2307.15818*, 2023.

Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Mottaghi, Priyam Parashar, et al. Partnr: A benchmark for planning and reasoning in embodied multi-agent tasks. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yun Chang, Nathan Hughes, Aaron Ray, and Luca Carlone. Hydra-Multi: Collaborative online construction of 3D scene graphs with multi-robot teams. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 10995–11002, 2023. doi: 10.1109/IROS55552.2023.10341838.

Guo Chen, Zhiqi Li, Shihao Wang, Jindong Jiang, Yicheng Liu, Lidong Lu, De-An Huang, Wonmin Byeon, Matthieu Le, Max Ehrlich, Tong Lu, Limin Wang, Bryan Catanzaro, Jan Kautz, Andrew Tao, Zhiding Yu, and Guilin Liu. Eagle 2.5: Boosting long-context post-training for frontier vision-language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=X2xLfqX24x.

Junting Chen, Checheng Yu, Xunzhe Zhou, Tianqi Xu, Yao Mu, Mengkang Hu, Wenqi Shao, Yikai Wang, Guohao Li, and Lin Shao. Emos: Embodiment-aware heterogeneous multi-robot operating system with llm agents. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Kaiyuan Chen, Shuangyu Xie, Zehan Ma, Pannag R Sanketi, and Ken Goldberg. Robo2vlm: Improving visual question answering using large-scale robot manipulation data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025b.

OX-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 1(2), 2023.

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

Frank Dellaert and Michael Kaess. Factor graphs for robot perception. *Foundations and Trends in Robotics*, 6(1–2):1–139, 2017. doi: 10.1561/2300000043.

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

Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2):230–244, 2022.

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.

Xiaofeng Gao, Qiaozi Gao, Ran Gong, Kaixiang Lin, Govind Thattai, and Gaurav S Sukhatme. Dialfred: Dialogue-enabled agents for embodied instruction following. *IEEE Robotics and Automation Letters*, 7(4):10049–10056, 2022.

Yuan Gao, Junfeng Chen, Xi Chen, Chongyang Wang, Junjie Hu, Fuqin Deng, and Tin Lun Lam. Asymmetric self-play-enabled intelligent heterogeneous multirobot catching system using deep multiagent reinforcement learning. *IEEE Transactions on Robotics*, 39(4):2603–2622, 2023.

Google DeepMind, April 2026. URL https://deepmind.google/models/gemma/gemma-4/.

Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010. doi: 10.1109/MITS.2010.939925.

Qiuyi Gu, Zhaocheng Ye, Jincheng Yu, Jiahao Tang, Tinghao Yi, Yuhan Dong, Jian Wang, Jinqiang Cui, Xinlei Chen, and Yu Wang. Mr-cographs: Communication-efficient multi-robot open-vocabulary mapping system via 3d scene graphs. *IEEE Robotics and Automation Letters*, 2025.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

Jiani Huang, Amish Sethi, Matthew Kuo, Mayank Keoliya, Neelay Velingker, JungHo Jung, Ser-Nam Lim, Ziyang Li, and Mayur Naik. ESCA: Contextualizing embodied agents via scene-graph generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3668–3678, 2015.

Yuanchen Ju, Yongyuan Liang, Yen-Jen Wang, Gireesh Nandiraju, Yuanliang Ju, Seungjae Lee, Qiao Gu, Elvis Hsieh, Furong Huang, and Koushil Sreenath. Momagraph: State-aware unified scene graphs with vision-language models for embodied task planning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=3eTr9dGwJv.

Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g^2o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation*, pp. 3607–3613, 2011. doi: 10.1109/ICRA.2011.5979949.

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

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

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Kehui Liu, Zixin Tang, Dong Wang, Zhigang Wang, Xuelong Li, and Bin Zhao. Coherent: Collaboration of heterogeneous multi-robot system with large language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10208–10214. IEEE, 2025a.

Yang Liu, Weixing Chen, Yongjie Bai, Xiaodan Liang, Guanbin Li, Wen Gao, and Liang Lin. Aligning cyber space with physical world: A comprehensive survey on embodied ai. *IEEE/ASME Transactions on Mechatronics*, 2025b.

Artem Lykov, Mikhail Litvinov, Mikhail Konenkov, Rinat Prochii, Nikita Burtsev, Ali Alridha Abdulkarim, Artem Bazhenov, Vladimir Berman, and Dzmitry Tsetserukou. Cognitivedog: Large multimodal model based system to translate vision and language into action of quadruped robot. In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 712–716, 2024.

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

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. IEEE, 2024.

Ryan J Marcotte, Xipeng Wang, Dhanvin Mehta, and Edwin Olson. Optimizing multi-robot communication under bandwidth constraints. *Autonomous Robots*, 44(1):43–55, 2020.

NVIDIA. Isaac sim, 2025. URL `https://docs.isaacsim.omniverse.nvidia.com/latest/index.html`.

OpenAI, March 2026. URL `https://openai.com/index/introducing-gpt-5-4/`.

Rolf Pfeifer and Fumiya Iida. Embodied artificial intelligence: Trends and challenges. *Lecture notes in computer science*, pp. 1–26, 2004.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=K3n5jPkrU6`.

Yiran Qin, Li Kang, Xiufeng Song, Zhenfei Yin, Xiaohong Liu, Xihui Liu, Ruimao Zhang, and Lei Bai. Robofactory: Exploring embodied agent collaboration with compositional constraints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10075–10085, 2025.

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

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026b. URL `https://qwen.ai/blog?id=qwen3.6-35b-a3b`.

Qincheng Sheng, Zhongxiang Zhou, Jinhao Li, Xiangyu Mi, Pingyu Xiang, Zhenghan Chen, Haocheng Xu, Shenhan Jia, Xiyang Wu, Yuxiang Cui, et al. A comprehensive review of humanoid robots. *SmartBot*, 1(1):e12008, 2025.

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*, pp. 10740–10749, 2020.

Siqi Song, Xuanbing Xie, Zonglin Li, Yuqiang Li, Shijie Wang, and Biqing Qi. Leveraging adaptive group negotiation for heterogeneous multi-robot collaboration with large language models. *arXiv preprint arXiv:2602.06967*, 2025.

Manycore Tech Inc. SpatialVerse Research Team. Interioragent: Interactive usd interior scenes for isaac sim-based simulation. `https://huggingface.co/datasets/spatialverse/InteriorAgent`, 2025.

Huajie Tan, Cheng Chi, Xiansheng Chen, Yuheng Ji, Zhongxia Zhao, Xiaoshuai Hao, Yaoxu Lyu, Mingyu Cao, Junkai Zhao, Huaihai Lyu, et al. Roboos-next: A unified memory-based framework for lifelong, scalable, and robust multi-robot collaboration. *arXiv preprint arXiv:2510.26536*, 2025a.

Huajie Tan, Xiaoshuai Hao, Cheng Chi, Minglan Lin, Yaoxu Lyu, Mingyu Cao, Dong Liang, Zhuo Chen, Mengsi Lyu, Cheng Peng, et al. Roboos: A hierarchical embodied framework for cross-embodiment and multi-agent collaboration. *arXiv preprint arXiv:2505.03673*, 2025b.

Gemma Team. Gemma 3 technical report, 2025.

Yulun Tian, Yun Chang, Fernando Herrera Arias, Carlos Nieto-Granda, Jonathan P. How, and Luca Carlone. Kimera-Multi: Robust, distributed, dense metric-semantic SLAM for multi-robot systems. *IEEE Transactions on Robotics*, 38(4):2022–2038, 2022. doi: 10.1109/TRO.2021.3137751.

Vishnunandan LN Venkatesh and Byung-Cheol Min. Zerocap: Zero-shot multi-robot context aware pattern formation via large language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 01–07. IEEE, 2025.

Qineng Wang, Wenlong Huang, Yu Zhou, Hang Yin, Tianwei Bao, Jianwen Lyu, Weiyu Liu, Ruohan Zhang, Jiajun Wu, Li Fei-Fei, et al. Enact: Evaluating embodied cognition with world modeling of egocentric interaction. *arXiv preprint arXiv:2511.20937*, 2025a.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025b.

Zelai Xu, Zhexuan Xu, Xiangmin Yi, Huining Yuan, Mo Guang, Kaiwen Long, Xinlei Chen, Yi Wu, Chao Yu, and Yu Wang. Vs-bench: Evaluating vlms for strategic abilities in multi-agent environments. *arXiv preprint arXiv:2506.02387*, 2025.

Ruibo Xue, Jiedan Tan, Fang Liu, Jingwen Tong, Taotao Wang, and Shuoyao Wang. From raw data to shared 3d semantics: Task-oriented communication for multi-robot collaboration. *arXiv preprint arXiv:2602.08624*, 2026.

Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 670–685, 2018.

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.

Chao Yu, Xinyi Yang, Jiaxuan Gao, Jiayu Chen, Yunfei Li, Jijia Liu, Yunfei Xiang, Ruixin Huang, Huazhong Yang, Yi Wu, et al. Asynchronous multi-agent reinforcement learning for efficient real-time multi-robot cooperative exploration. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pp. 1107–1115, 2023.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Yuanqian Zhao, et al. Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe. *arXiv preprint arXiv:2509.18154*, 2025.

Wenhao Yu, Jie Peng, Yueliang Ying, Sai Li, Jianmin Ji, and Yanyong Zhang. Mhrc: Closed-loop decentralized multi-heterogeneous robot collaboration with large language models. *arXiv preprint arXiv:2409.16030*, 2024.

Puzhen Yuan, Angyuan Ma, Yunchao Yao, Huaxiu Yao, Masayoshi Tomizuka, and Mingyu Ding. Remac: Self-reflective and self-evolving multi-agent collaboration for long-horizon robot manipulation. *arXiv preprint arXiv:2503.22122*, 2025.

Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5831–5840, 2018.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=EnXJfQqy0K.

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 *The Thirteenth International Conference on Learning Representations*, 2025.

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

APPENDIX

A OTHER DATASET DETAILS

A.1 DATA FORMAT

The dataset is stored in JSON format as timestep-level samples. Each sample corresponds to one agent at one timestep and contains the task context, observation, interaction history, and target outputs.

Observation. Each sample includes egocentric RGB observations, accompanied by metadata such as 3D bounding boxes, instance segmentation maps, and ground-truth scene graphs. These annotations support both raw multimodal evaluation and structured-representation studies.

History. Each agent receives the full interaction history up to timestep t, including its own previous actions and messages received from teammates. Messages are stored separately by sender, with consistent agent identifiers across the full episode.

Outputs. The target output consists of two fields: an action command and a communication message. The action is a high-level textual command designed to be executable by a downstream embodied controller, while the communication message is a concise natural language utterance used for coordination.

A.2 AGENT ROSTER AND PHYSICAL SPECIFICATIONS

Every scene in **SYNAPSE-Bench** features two heterogeneous robotic agents: a humanoid and a quadruped. The humanoid is instantiated as a Unitree G1 robot equipped with two Dex1 grippers, providing a total of 43 degrees of freedom (29 for full-body control and 7 per arm). It supports whole-body locomotion, and the dual grippers enable complex articulated manipulation. On the other hand, the quadruped is instantiated as a Unitree Go2 robot with 12 degrees of freedom. To facilitate coordination with the humanoid, it carries a small mounted carriage designed to transport lightweight objects such as a medical kit.

A.3 SCOPE OF THE COLLABORATION SETTING.

SYNAPSE-Bench conditions on a prescribed two-agent deployment: every episode specifies a humanoid–quadruped team whose observations, assigned roles, and available high-level actions are distributed across the two controllers. The benchmark therefore evaluates coordination quality within this deployment rather than the counterfactual benefit of adding a second agent. We do not claim that every scenario is physically impossible for a sufficiently capable single robot or that two-agent execution is always more efficient. Instead, the fixed team protocol *provides a controlled setting for measuring how decentralized VLM agents exchange information, track teammate progress, and select compatible actions under partial observability*.

A.4 TASK DESIGN AND COORDINATION STRUCTURE

Task-family construction. Table 5 summarizes the coordination structures in **SYNAPSE-Bench**. We define task families from the temporal organization of the reference trajectories. Every episode contains a humanoid loading action, and episodes are partitioned according to whether a humanoid door action occurs before loading, after loading, both before and after loading, or not at all. This definition yields four mutually exclusive families covering all 461 episodes. The families make the underlying division of labor explicit: the humanoid performs object and door manipulation, whereas the quadruped provides transport and must often wait until the humanoid makes its next action feasible. Figure 5 summarizes their empirical frequency and role specialization.

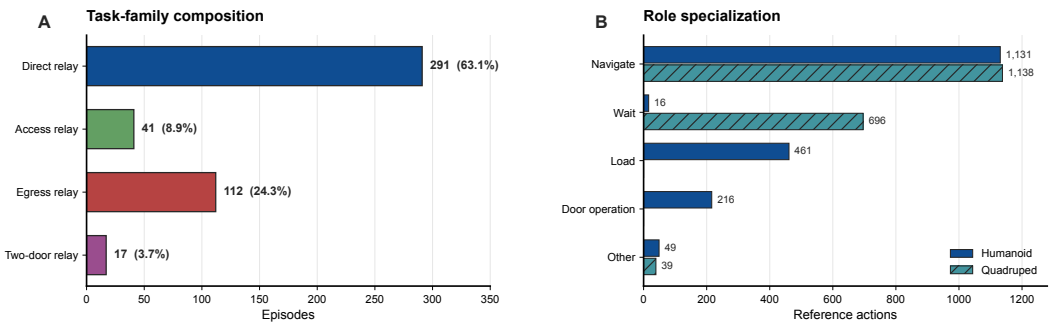


Figure 5: Quantitative overview of **SYNAPSE-Bench** task design. (A) Episode composition across the four coordination families. (B) Agent specialization in the reference actions. Loading aggregates `pick_up`/`pick`; door operations aggregate opening, closing, and explicit door-manipulation labels.

The most common case is the *direct relay* family (291 episodes, 63.1%), in which both agents can approach the target in parallel, after which the humanoid loads the medical kit and the quadruped delivers it. These episodes are compact, with a horizon of 3.29 ± 0.61 paired decision steps. The remaining 170 episodes add at least one explicit door dependency. In 41 *access-relay* episodes, the humanoid must first clear a door before the quadruped can approach and before loading can proceed. In 112 *egress-relay* episodes, loading occurs first, but delivery is gated by a later door operation. The 17 *two-door relay* episodes combine both constraints and are correspondingly longest (7.00 ± 1.46 steps, range 5–10). Thus, longer horizons are not merely longer navigation traces: they contain additional alternating dependencies between agents.

Coordination and information flow. The dependency column of Table 5 summarizes the direction in which one agent's completed action enables the other's next action. Direct relays typically begin with parallel navigation and then reduce to $H \rightarrow Q$: the humanoid's loading action enables quadruped transport. Door-conditioned families require repeated alternation. For example, $H \rightarrow Q \rightarrow H \rightarrow Q$ denotes the humanoid clearing a path, the quadruped advancing or reporting readiness, the humanoid loading or clearing the next constraint, and the quadruped completing delivery. The two-door family extends this pattern through both access and egress constraints.

These dependencies determine what must be communicated. The relevant messages are not generic dialogue: they convey target identity and location, door or obstacle state, permission to proceed, recipient readiness, loading completion, intended route or destination, and subgoal completion. This information is necessary because each agent is given an egocentric observation and a local belief state rather than a shared omniscient state. In the full benchmark, the per-step dependency fields contain 3,659 annotations. Of these, 2,791 name the quadruped as the dependent agent and 868 name the humanoid, quantitatively reflecting the asymmetric division of labor. The most frequent requirements concern loading the target object, opening a door, and establishing the other agent's readiness or completion state.

Figure 6 further shows how this burden changes with episode structure. Episodes of horizon three contain, on average, 1.13 wait actions and 0.36 request messages, whereas the longest observed episodes contain three wait actions and five requests. Although the longest-horizon bins contain fewer examples, the overall trend indicates that additional steps introduce coordination events rather than merely extending locomotion. Communication composition also changes across families: request acts account for approximately 11% of messages in the egress and two-door families, compared with roughly 5% in direct relays. Door-conditioned transport thus requires agents to communicate actionable state changes—particularly path blockage, clearance, and readiness—in addition to broadcasting progress.

Action-space asymmetry. Table 6 reports the high-level action annotations in the reference trajectories. Navigation is nearly balanced across agents (1,131 humanoid and 1,138 quadruped annotations), whereas synchronization and manipulation are strongly asymmetric. The quadruped produces 696 `wait` actions, compared with 16 for the humanoid, because it frequently waits for loading or

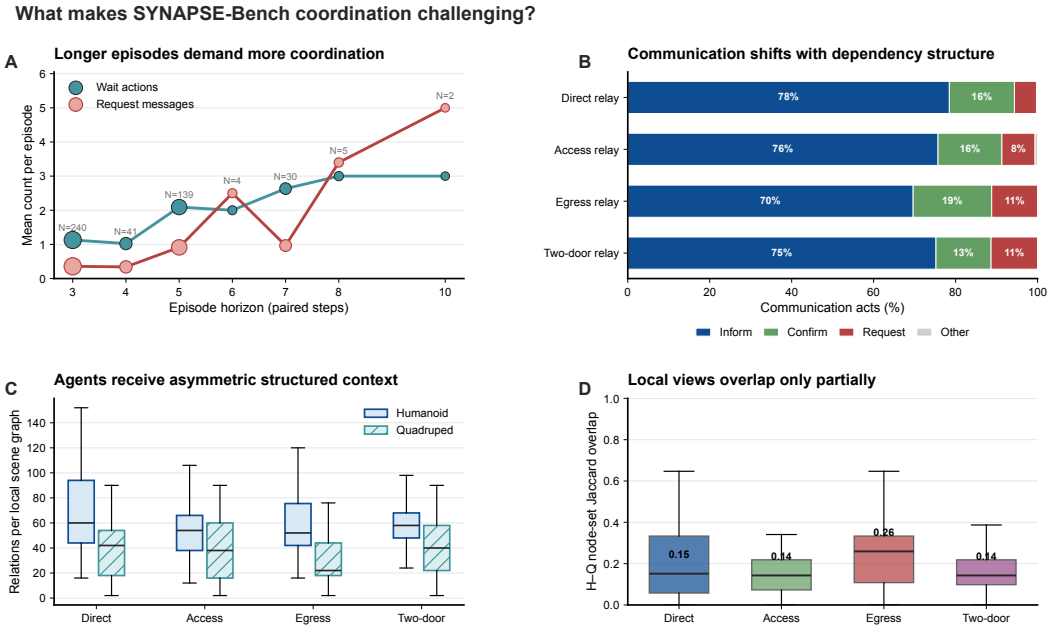


Figure 6: **Sources of coordination difficulty.** (A) Mean request messages and wait actions increase with episode horizon; marker size reflects the number of episodes and labels give exact sample sizes. (B) Communication-act composition by task family. Intent variants are consolidated into inform, confirm/acknowledge, request, and other acts. (C) Distribution of local scene-graph sizes for the two agents; boxes show the interquartile range and center lines show medians. (D) Pairwise Jaccard overlap between the node sets in simultaneous humanoid and quadruped scene graphs. The modest overlap quantifies the complementary local information available under partial observability.

door clearance. Conversely, all 461 loading annotations ("`pick_up`" and "`pick`") and all explicit opening annotations are assigned to the humanoid. This empirical separation is consistent with the capability metadata: the humanoid can manipulate objects and doors, whereas the quadruped is primarily a fast mobile carrier. Delivery is usually realized through `navigate`; consequently, the small count of the literal `deliver` label should not be interpreted as the number of delivery subgoals.

The label variants in Table 6 reflect surface-form variation in the annotations and should not be interpreted as distinct physical capabilities. For example, `pick_up` or `pick` describe the humanoid's loading operation. We preserve the variants in the count table for transparency while treating them as a single manipulation capability in the task-family analysis.

Sequential evaluation protocol. Sequential structure is represented by ordered humanoid–quadruped decision pairs, terminal indicators, local task descriptions, dependency annotations, and accumulated action/message history. During interactive evaluation, decision pairs are processed chronologically, and the actions and messages generated at one step become part of the agents' histories at the next step. History is advanced only after a successful paired decision. If either agent fails at a dependency-critical step, the episode terminates and subsequent subgoals are not credited. This protocol evaluates whether a model preserves the intended causal ordering across the complete trajectory, rather than scoring each decision as an independent sample.

Finally, each paired decision step specifies one local task for each agent, rather than a single global subgoal with a unique owner z_k. An exact cross-agent handoff count $\sum_k \mathbf{1}[z_k \neq z_{k+1}]$ is therefore not well-defined without introducing an additional ownership convention. We instead report the directly observable dependency patterns in Table 5, which capture the alternation of enabling actions without imposing a post hoc global-subgoal labeling.

Example of paired decisions and dependencies. Table 7 condenses selected decision pairs from a medical-kit reference trajectory. Intermediate pairs are omitted for brevity.

Table 5: Task-design coverage in the full **SYNAPSE-Bench** release. Families partition all 461 episodes by the position of humanoid door actions relative to loading. Horizon is measured in paired decision steps and shown as $\mathrm{mean}\pm\mathrm{std}$ (range). H and Q denote the humanoid and quadruped, respectively.

Task family	Example objective	H role	Q role	Dependency	Information exchanged	Horizon	N
Direct relay	Load the medical kit; deliver it to the bedroom.	Approach and load.	Approach, receive, and deliver.	Parallel; $H \rightarrow Q$	Target/location; readiness; load and delivery status	3.29 ± 0.61 (3–6)	291
Access relay	Open access; load and deliver the kit.	Clear access, retrieve, and load.	Await clearance, receive, and deliver.	$H \rightarrow Q \rightarrow H \rightarrow Q$	Door/path state; proceed signal; kit location; load status	6.02 ± 1.05 (5–8)	41
Egress relay	Load the kit; clear egress; deliver it.	Load, then open the blocking egress.	Receive, await clearance, and deliver.	$H \rightarrow Q \rightarrow H \rightarrow Q$	Load status; route; obstacle state; path-clear signal	4.91 ± 0.77 (3–8)	112
Two-door relay	Open access; load; clear egress; deliver.	Clear access, load, then clear egress.	Advance, receive, await clearance, deliver.	$H \rightarrow Q \rightarrow H$ $\downarrow$ $Q \leftarrow H \leftarrow Q$	Both door states; target/location; readiness; load and route status	7.00 ± 1.46 (5–10)	17

Table 6: Distribution of high-level action annotations by agent. Alternative spellings are shown separately when they occur as distinct labels.

Action label	Humanoid	Quadruped	Note
`navigate`	1131	1138	Both agents
`wait`	16	696	Mainly synchronization
`move`	4	25	Free-form targets
`pick_up/pick`	381/59/21	0	Label variants
`open/open_door`	196/17	0	Humanoid only
`close_door`	1	0	Humanoid only
`confirm/confirm_door_open`	33/1	0	Heterogeneous targets
`stay/assist`	9/2	0	Humanoid only
`manipulate/verify`	1/1	0	Humanoid only
`proceed/deliver`	0	6/4	Quadruped only
Other singleton Q labels	0	4	request, prepare, enter, place

Table 7: Selected paired decisions from one medical-kit reference trajectory. Intermediate decision pairs are omitted.

Stage	Humanoid	Quadruped
Approach	**Action:** Navigate to the medical kit. **Message:** Will open the kitchen door.	**Action:** Navigate to the kitchen door. **Message:** Requests that the door be opened.
Loading	**Action:** Pick up the medical kit. **Message:** Will load it onto the quadruped.	**Action:** Wait. **Message:** Waiting for loading.
Door access	**Action:** Navigate toward the bedroom. **Message:** Asks the quadruped to move aside.	**Action:** Move away from the bedroom door. **Message:** Clearing the path.

The stages are descriptive summaries, not additional scored subgoals. The evaluator scores the paired action–message outputs at successive decision steps. This reference trajectory illustrates one valid coordination sequence; another valid strategy may differ from it.

A.5 Scene Graph Annotation Construction

We construct scene graph annotations from the 3D simulation metadata provided by NVIDIA Isaac Sim (NVIDIA, 2025). Specifically, we leverage ground-truth 3D bounding boxes for all objects in each scene, including agents and interactable entities. These annotations provide precise spatial information (position, orientation, and extent) for all visible objects at each timestep.

Based on these 3D bounding boxes, we define a set of geometric rules to assign pairwise predicates between objects. In particular, we focus on relationships between the subject agents (humanoid and quadruped) and surrounding objects, as well as object–object relations when relevant. Predicates are determined using relative spatial configurations (e.g., left/right, in front/behind, proximity) computed from object centroids and bounding box overlaps.

To ensure annotation quality, all automatically generated scene graphs are further verified using an in-house annotation tool and human annotators. Annotators manually inspect and correct object relationships on a per-frame basis, resolving ambiguities caused by occlusion, viewpoint differences, or geometric edge cases. This process results in high-quality, temporally consistent scene graph annotations across the dataset.

A.6 Key Properties of the Benchmark

Multi-turn dependency. Tasks require temporally ordered reasoning: earlier actions modify the feasibility of later ones. For example, a quadruped may be unable to access an object until the humanoid opens a blocked path.

Cross-agent coupling. All episodes are specified as two-agent rollouts with role-conditioned subgoals. This design evaluates coordination within a fixed team and does not establish single-agent infeasibility.

Communication opportunity. Distributed viewpoints and role-conditioned actions create opportunities to exchange object locations, obstacle states, completed subgoals, and intended actions. The current benchmark compares communication interfaces but does not establish that explicit messages are necessary in every episode.

Partial observability. Agents operate from limited egocentric viewpoints and do not observe the full environment state. Critical information, such as object location, door state, or teammate progress, may be available to only one agent at a given timestep.

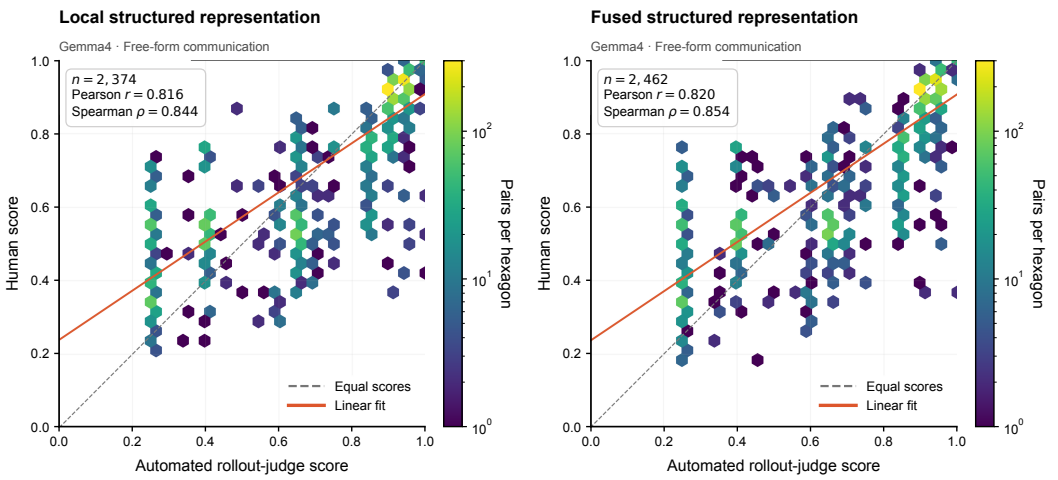
Interactive Loop. The dataset highlights an interactive loop between agents, where each continually adapts to the other's actions over multiple turns. The humanoid typically shapes the environment (e.g., manipulating objects or setting up conditions) while the quadruped responds by executing navigation-focused behaviors. This ongoing back-and-forth creates a dynamic coordination cycle, making the benchmark well-suited for studying both explicit communication and implicit collaboration in multi-turn interactions.

B EXPERIMENT DETAILS

Implementation. We evaluate multiple vision-language models (VLMs) in a decentralized setting, where each agent is controlled by an independent model instance. At each timestep t, agent i receives its local observation o_t^i (RGB frames), task description, interaction history, and optionally a structured belief representation (scene graph). Visual input is constructed by sampling up to 4 frames from the agent's egocentric video. All inputs are serialized into a unified prompt combining visual tokens with structured fields (observation, history, belief). The model produces two outputs: a physical action a_t^i and a communication message m_t^i. Actions are generated as free-form text and mapped to executable primitives via a predefined parser, while communication messages are appended to teammates' histories at the next timestep. For structured representations, we construct *local scene graphs* from each agent's observation and optionally provide a *fused scene graph* using GRAPHSTITCH. The fused graph is recomputed at each timestep and shared across agents.

Evaluation protocol. We perform zero-shot evaluation without task-specific fine-tuning. Each model is prompted with a fixed system instruction specifying the task, agent capabilities, and output format. Decoding uses temperature 0.2 during rollout, with a maximum context length of 32,768 tokens. Each evaluation involves either one VLM (homogeneous setting) or two VLMs (heterogeneous setting), together with a judge model (GPT-4o-mini) that verifies task completion and subgoal satisfaction. Episodes are executed in a synchronized multi-agent loop until task completion or a fixed horizon. We evaluate on 100 randomly sampled episodes. While the full dataset contains 461 episodes, we adopt a fixed evaluation subset to ensure consistent comparison across models and to control computational cost, particularly for large-scale VLM inference. The subset is sampled to preserve the distribution of tasks and environments, and we verify that results are stable across multiple random seeds. The remaining episodes are intended for extended evaluation, ablations, and future benchmarking. Open-source models are run on a cluster of 8×A100 (40GB) GPUs.

Correlation with Human Scores. Figure 7 compares the average score assigned by the automated rollout judge with the corresponding human score. We observe strong positive correlations in both representation settings. For local scene graphs, Pearson correlation is $r = 0.816$ and Spearman rank correlation is $\rho = 0.844$ across $2{,}374$ evaluated pairs. For fused scene graphs, the corresponding values are $r = 0.820$ and $\rho = 0.854$ across $2{,}462$ pairs. The similar coefficients indicate that graph fusion does not materially alter the relationship between automated and human scoring, while the high Spearman correlations show that the rollout judge largely preserves the ordering induced by human scores. These correlations demonstrate consistency with human evaluation, but they do not establish perfect score calibration. In particular, the fitted lines deviate from the equal-score diagonal near the ends of the score range, suggesting compression toward intermediate scores. We therefore interpret the results as evidence of correlation with human scores rather than treating the automated judge and human evaluator as interchangeable.



(a) (b)

Figure 7: Correlation between automated rollout-judge scores and human scores for Gemma4 with free-form communication under (a) local structured and (b) fused structured representations. Hexagon color denotes the number of evaluated agent pairs on a logarithmic scale. The dashed line indicates equal scores, and the solid line shows a linear fit.

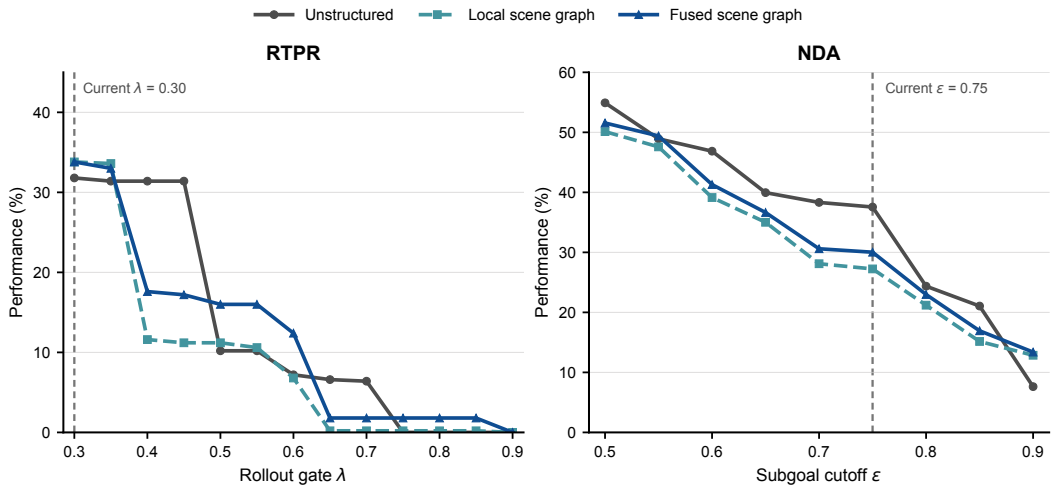


Figure 8: Gemma4 threshold sensitivity across five runs on the same 100 episodes in each representation setting. Left: RTPR as the rollout gate λ varies, with $\varepsilon = 0.75$. Right: NDA as the reporting cutoff ε varies, with $\lambda = 0.30$. Vertical dashed lines mark the values used in the reported experiments. The λ curve is a replay of stricter gates on the saved trajectories; it does not estimate the effect of gates below 0.30.

Sensitivity to rollout and reporting thresholds. The rollout judge assigns each agent an action score and a communication score in $[0, 1]$. Their arithmetic mean is the agent-level score s_i. At a paired step, the rollout continues only if both agents satisfy $s_i > \lambda$; the reported subgoal measure credits a reached pair if the mean of the two agent-level scores is at least ε. We use $\lambda = 0.30$ for rollout continuation and $\varepsilon = 0.75$ for subgoal reporting. These parameters serve different purposes: changing λ can truncate an episode and therefore changes RTPR, whereas changing ε only changes the offline NDA count.

Figure 8 sweeps both thresholds using the saved Gemma4 rollouts for unstructured, local-graph, and fused-graph inputs. At $\lambda = 0.35$, RTPR remains close to its value at 0.30 (31.4% versus 31.8% for unstructured inputs, 33.6% versus 33.8% for local graphs, and 33.0% versus 33.8% for fused

graphs). At $\lambda = 0.40$, however, local and fused RTPR fall to 11.6% and 17.6%. This discontinuity reflects the strict continuation rule and the discrete judge-score distribution: many paired steps have a minimum agent score of exactly 0.40 and fail when the gate reaches that value. Thus, the experiment is locally stable near the chosen rollout gate but is not insensitive to larger changes.

NDA also depends on its reporting cutoff. Raising ε from 0.75 to 0.80 changes the unstructured, local, and fused rates from 37.55%, 27.24%, and 30.02% to 24.36%, 21.20%, and 22.97%, respectively. At the reported thresholds, the fused-minus-local NDA difference is 2.78 percentage points; an episode-clustered bootstrap interval is $[-1.61, 7.18]$ percentage points (1,000 resamples of episode IDs, carrying all five runs and settings together). The interval includes zero, so these Gemma4 runs do not establish a clear fused-graph advantage. The sweep characterizes sensitivity to the stated thresholds; it is not an independent calibration of their optimality.

The λ replay is valid only for thresholds at least as strict as the original 0.30 gate, conditional on the saved model outputs and judge scores. A looser gate would require predictions for steps that were never generated after early stopping and would also alter subsequent interaction histories.

C Additional Discussion

Limitations. SYNAPSE-Bench has several limitations. (i) *Simulation gap.* The dataset is constructed in NVIDIA Isaac Sim, which provides a controlled approximation of real-world indoor environments but does not fully capture real-world perception, physics, or sensor noise. (ii) *Limited diversity.* Environments are restricted to a small set of indoor layouts and object configurations. This introduces selection bias and limits generalization to more diverse or unstructured settings. (iii) *Simplified agent roles.* The predefined humanoid manipulation and quadruped transport roles constrain the available coordination strategies. The results concern high-level coordination under capability asymmetry and do not establish challenges unique to humanoid–quadruped morphology. (iv) *Reference method scope.* GraphStitch is a lightweight fusion baseline and does not represent the full space of structured reasoning or learning-based fusion approaches. (v) *Communication necessity.* SYNAPSE-Bench evaluates decentralized agents with local observations and an explicit messaging channel, but the present experiments do not establish that messages are necessary in every episode. Agents may sometimes coordinate implicitly through observable actions or environmental changes. Moreover, we do not compare against a centralized single-agent policy with access to both agents' observations and action spaces. Consequently, our results should be interpreted as evaluating communication and representation under decentralized execution, not as proving the infeasibility of centralized or communication-free solutions.

Broader Impact. SYNAPSE-Bench may benefit assistive robotics, logistics, and search and rescue, but improved coordination also creates dual-use risks: autonomous teams or swarms could share observations and divide labor with less human oversight, including in surveillance, military, or other safety-critical applications. Benchmark performance should therefore not be treated as evidence of safe deployment; real-world systems require application-specific validation, human intervention mechanisms, auditable communication, and team-level fail-safes. Large-scale multi-agent VLM evaluation also consumes substantial GPU or proprietary-API resources, creating environmental costs and barriers for resource-constrained researchers. We reduce, but do not eliminate, these barriers by releasing the data, prompts, and code, and by supporting smaller and open-weight models. Finally, because the simulated data contain no demographic attributes, the benchmark cannot assess human-centered fairness; its limited environments and language conventions may also underrepresent users with different languages, disabilities, or assistive needs.

D GraphStitch Algorithm: The Reference Fusion Method

Motivation and design choice. Each agent observes only a partial scene, so its local graph is an incomplete and potentially inconsistent description of a common environment. The main fusion problem is therefore not to concatenate triplets, but to reconcile relational evidence expressed from different local views into a single shared spatial frame. GraphStitch draws on the constraint-optimization view used in graph-based SLAM and factor-graph inference, where pairwise spatial observations are represented as graph edges and a globally consistent configuration is recovered by minimizing their weighted residuals (Grisetti et al., 2010; Kümmerle et al., 2011; Dellaert &

Kaess, 2017). Its alignment of local graphs in a shared reference frame is also related to multi-robot metric-semantic mapping and collaborative 3D scene-graph construction (Tian et al., 2022; Chang et al., 2023). Unlike these full metric mapping systems, however, GRAPHSTITCH operates on lightweight qualitative relations and is intended as a reference fusion module rather than a general SLAM system.

We choose GRAPHSTITCH as the benchmark reference because it provides the minimal machinery needed to reconcile partial structured observations: node alignment identifies shared entities, spatial relations become explicit geometric constraints, and anchor normalization removes frame ambiguity. The method is deterministic, training-free, interpretable, and independent of the downstream VLM. Consequently, performance differences between the local and fused settings primarily reflect the effect of information aggregation rather than an additional learned fusion model. Its low computational overhead also allows the shared graph to be recomputed at every timestep.

Interpretation. After node alignment, let $x_v \in \mathbb{R}^d$ denote the latent position of canonical node v. A directional relation supplies a noisy relative-position measurement $x_o - x_s \approx \Delta(r)$, whereas a distance relation constrains $\|x_o - x_s\|_2$ to an interval $[d^-(r), d^+(r)]$. Thus, GRAPHSTITCH can be viewed as a weighted relational-consensus estimator:

$$\widehat{X} = \arg\min_{X:\, x_{v_c}=0} \sum_{c\in\mathcal{C}} w_c\, \ell_c(X), \tag{2}$$

where w_c controls the reliability of constraint c and ℓ_c penalizes its violation. For example, directional constraints may use $\ell_c(X) = \|(x_o - x_s) - \Delta(r)\|_2^2$, while distance constraints may use a hinge penalty that is zero inside the admissible interval. Under independent Gaussian noise on directional offsets, the directional part of Eq. 2 is a maximum-likelihood estimator, with w_c proportional to inverse noise variance.

The anchor constraint $x_{v_c} = 0$ is not merely an implementation detail. Because all input relations depend only on relative positions, translating every node by the same vector leaves the objective unchanged. Anchoring fixes this gauge freedom. In the directional-only case, with fixed node correspondences, positive weights, and a connected constraint graph, the quadratic objective has a unique anchored minimizer: its Hessian is a weighted graph Laplacian, whose only unanchored null direction is global translation. This also makes the role of cross-agent overlap explicit. Shared aligned nodes connect otherwise local constraint graphs; without such overlap, their relative placement is unidentifiable. Distance constraints extend the model to coarse spatial evidence but can make the full objective non-convex, so the method should be understood as a transparent reference estimator rather than a globally optimal mapping system.

This formulation explains why GRAPHSTITCH is suitable for the benchmark. It is permutation-invariant to the order of input graphs, supports partial overlap, exposes contradictory evidence through constraint residuals, and produces a common symbolic interface for every evaluated model. At the same time, it makes limited assumptions and introduces no task-specific training. More sophisticated approaches, including probabilistic data association, robust pose-graph optimization, or learned graph fusion, can replace it without changing the benchmark protocol.

Algorithm: GRAPHSTITCH

Require: Local scene graphs $\mathcal{G}^A = (\mathcal{V}^A, \mathcal{E}^A)$ and $\mathcal{G}^B = (\mathcal{V}^B, \mathcal{E}^B)$, with edges (s, o, r)
Require: Directional relations $\mathcal{R}_{\text{dir}}$, distance relations $\mathcal{R}_{\text{dist}}$, anchor node v_c
Ensure: Fused scene graph $\mathcal{G}^F = (\mathcal{V}^F, \mathcal{E}^F)$

Align nodes.
Compute correspondence map $\pi : \mathcal{V}^B \to \mathcal{V}^A \cup \{\text{NEW}\}$ using node labels and spatial context
Canonicalize nodes: $\mathcal{V}^F \leftarrow \mathcal{V}^A \cup \{\pi(v) : v \in \mathcal{V}^B\}$
Rewrite edges in $\mathcal{E}^B$ according to π

Build spatial constraints.
Initialize constraint set $\mathcal{C} \leftarrow \emptyset$
for all $(s, o, r) \in \mathcal{E}^A \cup \mathcal{E}^B$ **do**

if $r \in \mathcal{R}_{\text{dir}}$ **then**
    Add $(o, s, \Delta(r))$ to $\mathcal{C}$, where $\mathbf{x}_o - \mathbf{x}_s \approx \Delta(r)$
else if $r \in \mathcal{R}_{\text{dist}}$ **then**
    Add $(o, s, d^-(r), d^+(r))$ to $\mathcal{C}$, where $d^-(r) \leq \|\mathbf{x}_o - \mathbf{x}_s\|_2 \leq d^+(r)$
end if
end for

Normalize to the anchor frame.
Set $\mathbf{x}_{v_c} \leftarrow \mathbf{0}$
Estimate node positions by minimizing constraint violations:

$$\{\mathbf{x}_v\}_{v \in \mathcal{V}^F} \leftarrow \arg\min_{\{\mathbf{x}_v\}} \sum_{c \in \mathcal{C}} w_c \, \ell(c)$$

where $\ell(c)$ is the violation penalty of directional or distance constraint c

Reconstruct fused relations.
Initialize $\mathcal{E}^F \leftarrow \emptyset$
for all ordered node pairs $(s, o) \in \mathcal{V}^F \times \mathcal{V}^F$, $s \neq o$ **do**
    Compute $\mathbf{d}_{s,o} \leftarrow \mathbf{x}_o - \mathbf{x}_s$
    Infer relation $\hat{r} \leftarrow \text{Relabel}(\mathbf{d}_{s,o})$
    Add $(s, o, \hat{r})$ to $\mathcal{E}^F$
end for

return $\mathcal{G}^F = (\mathcal{V}^F, \mathcal{E}^F)$

E PROMPTS

System Prompt

```
You are a cooperative multi-agent decision model operating in an indoor environment.
Your task is to generate the "output" field for the given input.
Environment assumptions:
- Indoor space with rooms, doors, and objects
- Agents include:
  - Humanoid robot: can navigate, manipulate objects, open doors
  - Quadruped robot: fast navigation, cannot open doors or manipulate complex objects
Decision Rules:
- Use observation, history, belief, and teammate_model
- Respect physical feasibility (no impossible actions)
- Resolve dependencies in coordination.dependencies
- If a teammate is blocked and you can help,  prioritize helping
- If you are blocked,  communicate clearly
- Always include both:
  - action
  - communication
Action Requirements:
- Must be executable given your capabilities
- Must be relevant to the task
Communication Requirements:
- Must support coordination (inform, request, confirm)
- Must be concise and actionable
Output Format (STRICT):
Return ONLY this JSON:
{
   "action": {
        "type": string,
        "target": string
    },
    "communication": {
        "target": string,
        "intent": string,
        "content": string
    }
}
```

```
Do NOT:
- Modify or repeat the input
- Add extra fields
- Add explanations or text outside JSON
Now generate the output for the following input:
```

GPT-4o-mini as a Judge

```
You are a strict evaluator. You will score the action taken and the effectiveness of communication, and
the score will follow the following JSON format.

Ground truth:
{gt}

Prediction:
{pred}


Strictly follow this JSON format to give the score:
{{
    "action_score": number,
    "communication_score": number
}}

Each score ranges from 0 to 1 based on correctness, with the resolution of 0.1
```