SHAPE: Scaling Collaborative Planning in LLM-Powered Heterogeneous Multi-Robot Systems

Hanqing Yang[1], Zixuan Weng[1], Junjia Guo[2], Jingdi Chen[2], and Carlee Joe-Wong[1]

***Abstract*— Planning in heterogeneous robot teams requires agents to coordinate work allocation based on task requirements and their capability compositions. While adding robots to the team may increase their ability to accomplish more complex tasks or complete multiple tasks faster, coordinating their plans may introduce overhead without improving task performance. Understanding this trade-off requires distinguishing opportunities within a task from how effectively a planning method uses them. In this work, we systematically study how collaboration scales across task configurations, collaboration patterns, and team compositions. We characterize tasks with two measures derived from typed task graphs: parallel participation, which captures opportunities for additional robots to work concurrently; and heterogeneous participation, which captures the need for contributions from different robot types. These measures guide the construction of 8 embodied tasks across four configurations and two scenarios. We propose SHAPE (SHared Asynchronous Planning with Evolving tasks), which supports adaptive, asynchronous multi-robot planning through a shared, evolving, agent-built task graph and selective notifications. The graph provides a shared task-level representation that agents update as they plan and execute. We evaluate teams of 3, 9, and 15 robots, comparing SHAPE with Individual, Centralized, and Chain in task performance and coordination overhead. Our results highlight a trade-off: shared task understanding and timely agreement support effective collaboration, but synchronization costs can limit scaling. SHAPE achieves higher average completion with lower overhead across scales. A shared-channel ablation examines the role of structured task state, and a case study assesses the effect of grouping robots by type.**

I. Introduction

Heterogeneous multi-robot systems can distribute work across a team and combine capabilities to accomplish tasks beyond the ability of any single robot [1]–[3]. LLMs offer flexible task planning and coordination for such teams [4]–[6]. However, adding more robots does not necessarily improve task performance and may instead increase collaboration overhead [7], [8]. Scaling collaborative planning in LLM-powered heterogeneous robot teams presents two challenges: *evaluating* the capacity additional robots offer and how effectively it is used, and *designing* mechanisms to coordinate agents' plans without excessive overhead.

Evaluation of heterogeneous collaboration. Heterogeneous collaboration is shaped by interacting factors whose effects are difficult to disentangle: task demands, team composition, and collaboration patterns. First, tasks differ in how capability requirements are distributed across sequential and parallel work [10]. Additional robots may have little useful work to undertake or lack required capabilities; independent task branches may compete for eligible robots; and dependency constraints may prevent concurrent progress, causing agents to idle. Second, team composition, the distribution and possible combinations of robot types, determines whether the required capabilities and sufficient parallel capacity are available. More robots of the same type may increase parallel capacity; robots with different capabilities may enable complex work requiring joint capabilities. Third, collaboration patterns determine how agents collaborate through roles and communication, potentially introducing duplicate effort, waiting, and synchronization costs. Task completion rate alone can hardly reveal whether scaling benefits arise from task opportunities, team capabilities, or effective coordination; evaluation must consider both task performance and collaboration overhead [10]–[12]. Existing scaling studies examine reasoning and tool-use benchmarks [12]–[15] or simplified embodied settings [16], [17]. However, it remains unclear how task demands and robot-type composition shape the benefits of scaling under embodied constraints.

To support interpretable comparisons of embodied heterogeneous robot collaboration, we represent each task as a typed directed acyclic graph (DAG) and characterize its demands using two quantifiers: parallel participation, which captures how much work can proceed concurrently; and heterogeneous participation, which captures demands of different robot types along sequential paths and within layers. These measures guide the construction of tasks with controlled

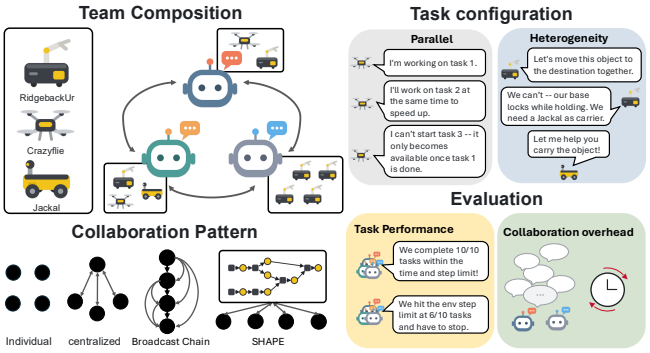


Fig. 1. **Overview of scaling collaborative planning in heterogeneous robot teams.** We vary team composition, task demands, and collaboration patterns to understand the benefits and collaboration costs of adding robots. Tasks vary in parallel and heterogeneous participation (Sec. IV-A). We compare Individual, Centralized, Chain, and SHAPE in terms of task performance and collaboration overhead (Secs. III-D and IV-C).

Conference on Robot Learning (CoRL 2026) Workshop: Learning Effective Abstractions for Planning (LEAP).

[1]Hanqing Yang, Zixuan Weng, and Carlee Joe-Wong are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA. Contact: hanqing3@andrew.cmu.edu.

[2]Junjia Guo and Jingdi Chen are with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721, USA.

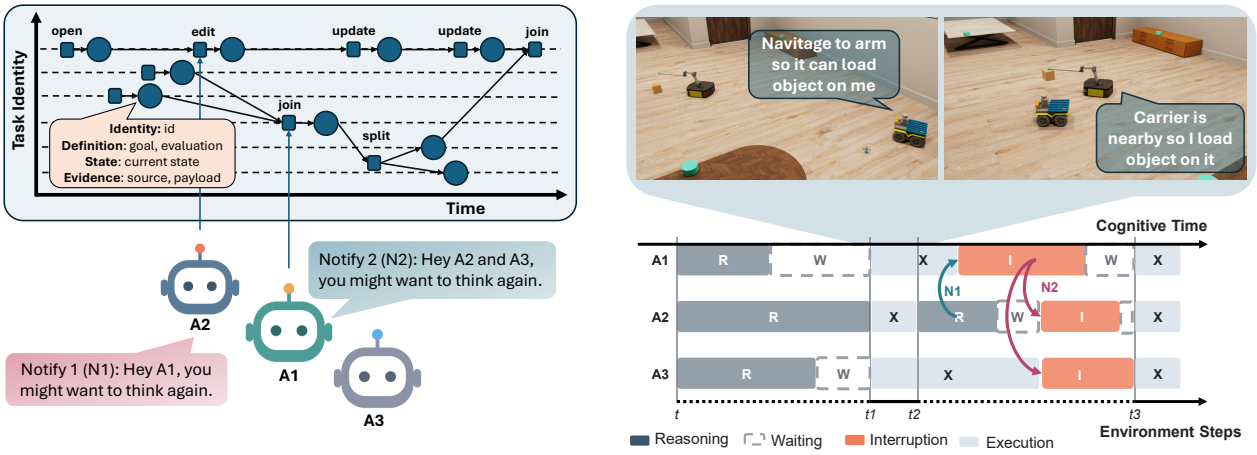


Fig. 2. **Asynchronous multi-robot planning with SHAPE.** Agents create and update a shared task representation to track task progress, revise their own plans, and coordinate work within and across robot teams. Selective notifications let agents interrupt one another to synchronize promptly when needed with minimal overhead. **Left:** the agent-built Task Activity Graph (TAG) represents evolving tasks through task versions (circles) and actions (squares); dashed lanes denote persistent task identities [9]. **Right:** interactions among agents and with the environment are tracked through transitions among four stages: reasoning (R), waiting (W), execution (X), and interruption (I) [10].

demands in BEHAVIOR [18], [19], a benchmark for embodied household activities in simulated 3D environments. Robots must coordinate under spatial and capability constraints: adding robots may increase collision risk and complicate navigation and co-location, while moving objects requires eligible robots to be within reach of target objects or one another. These requirements extend the evaluation beyond reasoning and message exchange to account for embodied uncertainties, allowing us to evaluate task performance and collaboration overhead across team sizes, compositions, and collaboration patterns.

Heterogeneous collaboration across scales. Scaling heterogeneous collaboration requires balancing shared task understanding and timely agreement against synchronization costs. Insufficient coordination can cause duplicate effort or incompatible plans, while excessive synchronization can introduce waiting and repeated interruptions. Effective collaboration, therefore, requires distributing work and adapting collaboration patterns as task, environment, and agent states evolve. Traditional approaches often rely on predefined protocols or policies trained for specific domains and team configurations [20], [21], while LLMs offer flexible coordination through reasoning, communication, and tool use [22], [23]. Recent embodied studies explore different collaboration patterns in LLM-powered multi-robot systems, including structured dialogue among robots and centralized planning for task decomposition, allocation, and feedback-driven multi-stage execution [4]–[6], [24]–[26]. These patterns coordinate agents by exchanging messages about the current state through communication structures; however, communication overhead can increase as teams grow, while fixed communication structures may also limit adaptation to changing coordination needs as immediate task demands change.

We introduce SHAPE (SHared Asynchronous Planning with Evolving tasks), which builds on DIG-TAG [9] to support multi-robot planning by letting agents asynchronously create and revise a shared, evolving, agent-built task graph and selectively notify others. This makes current task state structurally visible while allowing agents to choose when further coordination is needed. SHAPE achieves higher average completion with lower collaboration overhead across tasks than the baselines, motivating collaboration mechanisms that adapt to evolving task demands while balancing the benefits of shared task understanding, diverse perspectives, and timely agreement with collaboration costs.

Our contributions. We make three main contributions. **(1) Evaluating heterogeneous collaboration.** We characterize task demands through parallel and heterogeneous participation, enabling interpretable scaling comparisons and the controlled construction of eight tasks spanning four configurations across two scenarios. **(2) SHAPE for scalable collaboration.** We propose an asynchronous planning method in which LLM agents coordinate robot teams through a shared, evolving, agent-built task graph. Each agent plans for one team, updates shared task state, and selectively notifies others. **(3) Scaling study.** We evaluate teams of $N \in \{3, 9, 15\}$ robots, with equal numbers of RidgebackUr mobile manipulators, Jackal mobile robots, and Crazyflie quadrotors. We compare SHAPE with Individual, Centralized, and Chain in task performance and collaboration overhead, including a shared-channel ablation and a case study of teams grouped by type.

The remainder of this paper is organized as follows. Section II reviews multi-robot collaboration, agent scaling, and collaborative task design. Section III presents our asynchronous planning method, Section IV introduces our task characterization and evaluation methodology for heterogeneous robot teams, and Section V reports the results.

II. Related Work

We outline the relationship between our scaling study and asynchronous planning method with prior work on methods for LLM-based robot collaboration, prior scaling studies of

LLM agent systems, and existing evaluation frameworks for multi-agent collaboration in general.

A. LLM-Based Multi-Robot Collaboration Methods

LLM-powered multi-robot systems may follow centralized, decentralized, or hybrid collaboration patterns [8]. Centralized methods such as SMART-LLM and COHERENT [5], [6] use an orchestrator to coordinate robots directly or through robot agents. CoELA, RoCo, and DAMCS [4], [24], [27] use decentralized agents that coordinate with one another. Continual planning revises plans as the world changes [28]. LiP-LLM [29] optimizes allocation over an LLM-generated dependency graph, while OSDAG [30] schedules tasks online over a graph fixed before execution. In our method, each LLM agent plans for a robot team and coordinates asynchronously with other agents through a shared, evolving, evidence-grounded task graph and selective notifications, enabling agents to adopt more adaptive and task-specific collaboration patterns between robots and across teams.

B. Scaling LLM Agent Systems

Increasing the number of LLM agents in a given system can improve performance but also increase collaboration overhead. LLM-agent scaling is commonly evaluated on reasoning, mathematics, coding, and browser-, tool-, or terminal-use tasks [12]–[15]. These studies show that scaling effects depend on task structure, agent capability, and collaboration pattern; and that coordination overhead may offset the benefits of larger teams. However, most of these studies use non-embodied tasks, often with homogeneous agents, and offer limited insight into planning and coordination under heterogeneous robot capabilities and embodied constraints. Embodied work evaluates different collaboration patterns as robot count grows in simplified 2D scenarios and a 3D manipulation task [16], while CUBE scales a homogeneous agent population together with environment size and workload [10], [17]. We evaluate scaling across heterogeneous robot types in two realistic scenarios within BEHAVIOR.

C. Tasks and Evaluation for Multi-Agent Collaboration

Classical formulations of multi-robot task allocation and their heterogeneous extensions characterize capabilities, coalitions, task dependencies, decomposability, and coalition-size flexibility [2], [31], [32]. Interactive and embodied benchmarks study these requirements in different settings. For example, Collab-Overcooked gives two agents asymmetric resources, actions, and task knowledge [11], while Moving Out studies physically constrained two-agent collaboration [33]. RoCo uses heterogeneous teams of two or three robots [4], and CUBE requires agents to push blocks in a grid world under joint capability constraints [17]. COOP^2 [10] further formalizes embodied collaboration requirements to make the collaboration process evaluable. BDDL specifies the goal conditions used to evaluate task completion [18], [19], which the Embodied Agent Interface uses for evaluating single-agent embodied decision-making [34]. We quantify task demands for heterogeneous robot teams and design our scaling evaluation tasks in BEHAVIOR using BDDL.

III. Multi-Robot Planning and Collaboration

Our method supports asynchronous multi-robot planning through a shared, evolving, agent-built task graph. Agents plan for their own robot teams, create and revise tasks, record progress and evidence, and selectively notify others. We build on the DIG-TAG representation to define the shared task graph [9] and the COOP^2 interaction loop as a framework for asynchronous agent collaboration [10].

A. Shared Task State

We instantiate DIG-TAG as $DT = (G_D, G_T, Z, U)$, coupling the *Dynamic Interaction Graph* (DIG) G_D, the *Task Activity Graph* (TAG) G_T, environment state Z, and tool set U. The two graphs are

$$G_D = (E, H, L_D), \qquad G_T = (Q, A, \Phi, L_T). \tag{1}$$

In the DIG, E contains information events, H LLM-agent activations, and L_D their information-flow edges. Each activation records its available inputs, ordered tool calls, and returned events. In the TAG, Q contains task versions, A task actions (shown in Table I), L_T their relationships, and Φ evidence attachments.

TABLE I

Task, evidence, and notification tools with agent-supplied arguments.

Tool call	Description
$\mathsf{OPEN}(\delta, \sigma)$	Create a task with a fresh identity.
$\mathsf{EDIT}(q, \delta')$	Revise a goal or evaluation rule.
$\mathsf{UPDATE}(q, \sigma')$	Update the current state.
$\mathsf{SPLIT}(q, \{(k_j, \delta_j, \sigma_j)\}_{j=1}^{m})$	Create subtasks or branch out.
$\mathsf{JOIN}(\{q_j\}_{j=1}^{m}, k', \delta', \sigma')$	Combine input task versions.
$\mathsf{CLOSE}(k)$	Mark the identity as closed.
$\mathsf{ATTACH}(s, x, q)$	Attach evidence to task version q.
$\mathsf{NOTIFY}(J_j)$	Interrupt agents in J_j.

Agents create and revise tasks using the DIG-TAG tools (Table I). Each task version $q = (k, \delta, \sigma) \in Q$ records a persistent identity k, a specification δ containing its goal and evaluation rule, and a state σ. EDIT and UPDATE revise δ and σ by appending new versions with the same task identity. An attachment $\phi = (s, x, q) \in \Phi$ links a source s and payload x to an exact task version q. Agents attach evidence to task versions to indicate task status and inform subsequent decisions. For example, evidence may come from BDDL evaluation results or agent-generated messages.

B. Asynchronous Planning and Execution

Let J index the LLM agents. The Multi-Agent Environment Interaction Loop (MAEIL) from COOP^2 [10] tracks the agents asynchronously and separates cognitive time $\hat{t}$ from environment steps t. For each LLM agent $j \in J$, the trace records a stage $q_{j,\hat{t}} \in \{\mathrm{R}, \mathrm{W}, \mathrm{X}, \mathrm{I}\}$, denoting reasoning, waiting, execution, or interruption stages, with duration $\Delta\hat{t}_{j,\hat{t}}$.

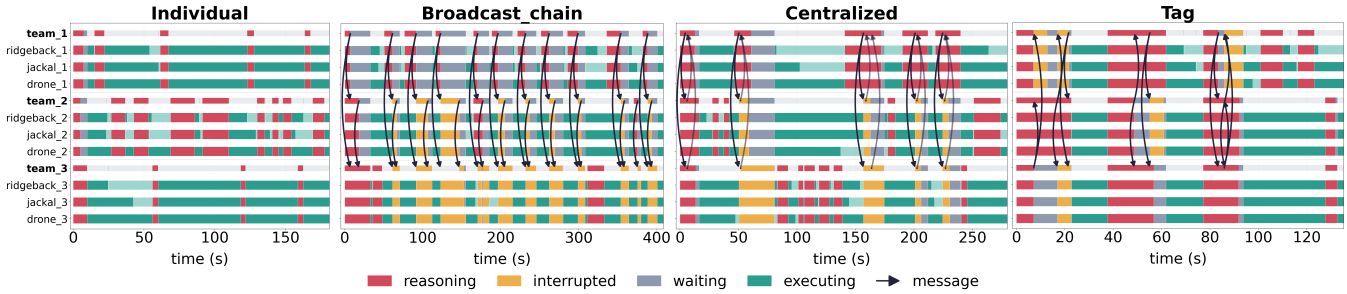


Fig. 3. **Interaction traces make collaboration overhead visible during planning and execution.** Each LLM agent plans its robot team's actions and may exchange messages with other agents. Rows show team planning and robot activity. Transitions among four stages capture interactions among agents and with the environment. Colors distinguish reasoning, waiting, execution, and interruption periods; arrows indicate messages between agents.

C. Agent collaboration with SHAPE

SHAPE (SHared Asynchronous Planning with Evolving tasks) enables asynchronous collaboration through a shared task graph [9]. Each agent can observe the task graph, so at each activation, agent $j \in J$ runs a tool-calling loop to update its plan using environment observations $o_{E,j}$ and task observations $o_{G_T,j}$. The task observations $o_{G_T,j}$ contain the latest versions of open tasks, agent work histories, and structural relationships among tasks. Each agent j has a notification budget $b_j \in \mathbb{Z}_{\geq 0}$, which resets after each environment step. Within this budget, the agent may call `notify` to send a notification message containing the sender's name to each selected recipient, causing the recipient to enter the interruption stage I and giving it a chance to revise its decisions. The environment advances only when all agents have actions ready for the next step. For each planning decision, agent j produces a tuple with three components:

$$(u_{T,j}, \hat{\pi}_j, J_j) \leftarrow \text{LLM}(o_{E,j}, o_{G_T,j}). \tag{2}$$

Here, $u_{T,j}$ specifies a TAG tool (Table I) and its arguments, $\hat{\pi}_j$ is the proposed execution plan, and $J_j \subseteq J \setminus \{j\}$ contains the selected recipients. Both $u_{T,j}$ and J_j may be empty, and no notification budget is consumed when $J_j = \varnothing$. The TAG effect is realized before `notify` is called.

D. Robot Teams and Collaboration Patterns

Let $J = \{0, \ldots, M-1\}$ and $K = \{0, \ldots, N-1\}$ denote the index sets of M LLM agents and N robots, respectively. Each agent $j \in J$ is responsible for one robot team $K_j \subseteq K$, with the teams partitioning K. It generates a plan for each robot $k \in K_j$ and, unless interrupted, resumes planning once all robots in its team complete their current plans.

Collaboration patterns can affect performance and scalability [12], [16]. We compare our method with three representative patterns. In *Individual*, agents plan independently without communicating. In *Centralized*, agent $0 \in J$ also serves as coordinator, and other agents exchange messages with it. In *Chain*, agents communicate in index order: each considers earlier messages, broadcasts its own message to all following agents, and then plans for its team. In our method, agent j selects a recipient subset $J_j \subseteq J \setminus \{j\}$ for each `notify` call.

$$\begin{aligned} E_{\text{ind}} &= \emptyset, & E_{\text{cen}} &= \{(0,j),(j,0) : j \neq 0\}, \\ E_{\text{chain}} &= \{(j,j') : j < j'\}, & E_{\text{ours}} &= \{(j,j') : j' \in J_j\}. \end{aligned} \tag{3}$$

IV. Evaluating Heterogeneous Multi-Robot Collaboration

We study how heterogeneous robot collaboration changes with task collaboration demands, team size and composition, and collaboration method. We use BEHAVIOR's BDDL language to specify task initial and goal conditions and OmniGibson for simulation and robot execution. Team size and composition determine available capabilities and capacity, while task demands determine how many robots can contribute and which capabilities are needed. The collaboration method determines how effectively the team can use these resources without excessive coordination overhead.

A. Quantifying Task Demands for Heterogeneous Teams

To systematically evaluate collaboration in heterogeneous robot teams, we characterize tasks using two measures: *parallel participation* $P(T)$, which captures opportunities to use larger teams, and *heterogeneous participation* $C(T)$, which captures the need for different robot capabilities. Crossing low and high levels of these measures yields four task configurations, each instantiated in two embodied scenarios, for 8 tasks in total.

Typed evaluation task graph. Let R be the set of robot types. We represent each task as a typed DAG $T = (V, E, \rho)$ for evaluation. Each node $v \in V$ represents a task goal, and $\rho_v \subseteq R$ contains the robot types eligible to complete it. Each directed edge $(u, v) \in E$ represents a dependency, such that the completion of v is counted only after u has been completed.

We instantiate the typed DAG T as evaluation criteria in BEHAVIOR and translate it into a task description containing task context and completion conditions. Based on this description, agents develop their own plans and use the agent-generated TAG G_T to maintain and revise a shared understanding of tasks and progress during execution.

Two quantifiers. (1) Parallel Participation $P(T)$ estimates the average amount of task work that can be performed concurrently, assuming sufficient eligible robots and resources. To construct dependency layers, we add a dummy source



Fig. 4. Example trajectory for transporting a light object. The RidgebackUr manipulator first navigates within reach of the object, and the Jackal carrier moves alongside it. The manipulator then grasps the object and loads it onto the waiting carrier. The manipulator cannot navigate while holding an object. After transport, it unloads the object and places it on the target. Each navigation spans multiple environment steps, with duration proportional to distance.

TABLE II

SYMBOLIC TOOLS. R, J, AND C DENOTE THE RIDGEBACKUR MANIPULATOR, JACKAL CARRIER, AND CRAZYFLIE QUADROTOR. STEPS DENOTES EACH TOOL CALL'S DURATION IN ENVIRONMENT STEPS. MANIPULATION REQUIRES TARGETS WITHIN DISTANCE r, REACHED VIA `NAVIGATE_TO` WHEN NEEDED. FAILED ACTIONS RETURN A TYPED ERROR CODE INSTEAD OF THE EXPECTED EFFECT.

Action	Argument	Steps	R	J	C	Expected effect
`navigate_to`	`object_id` / `robot_id`	$10d$	✓	✓	✓	Teleport near the target, carrying held and loaded objects; 10 steps per meter of travel distance d.
`grasp`	`object_id` (light)	1	✓	×	✓	Pick up an object into the hand; distance $\leq r$.
	`object_id` (heavy)	1	✓	×	×	
`place_on_top`	`object_id` (surface)	1	✓	×	✓	Put the held object on the target's surface; distance $\leq r$.
`load_onto`	`robot_id` (J only)	1	✓	×	✓	Move the held object onto the carrier's back; distance $\leq r$.
`unload_from`	`robot_id` (J only, light cargo)	1	✓	×	✓	Take the object riding on the carrier into the hand;
	`robot_id` (J only, heavy cargo)	1	✓	×	×	distance $\leq r$.
`wait`	$t \in [1, 600]$	t	✓	✓	✓	Hold the current configuration for t steps.

node v_0, representing a starter task with no meaningful work, and connect it to every task node that originally has no predecessor. Let $\ell(v)$ denote the number of edges in the longest directed path from v_0 to v, with $\ell(v_0) = 0$. The number of dependency layers is $K = \max_{v \in V} \ell(v)$, and the width of layer k is $b_k = |\{v \in V : \ell(v) = k\}|$. **(2) Heterogeneous Participation** $C(T)$ measures the average fraction of robot types required by each subtask. Let R denote the set of robot types in the team and $\rho_e \subseteq R$ the types required by edge $e \in E$. The fraction required by edge e is $\delta(e) = |\rho_e|/|R|$.

$$P(T) = \frac{1}{K} \sum_{k=1}^{K} b_k, \qquad C(T) = \frac{1}{|E|} \sum_{e \in E} \delta(e). \tag{4}$$

Task Configurations. We cross low and high levels of parallel participation $P(T)$ and heterogeneous participation $C(T)$ to form the four configurations shown. The first and second letters denote the levels of $P(T)$ and $C(T)$, respectively. All tasks have a similar total workload, with the same numbers of nodes $|V|$ and edges $|E|$.

	LL	LH	HL	HH
$P(T)$	1	1	5	5
$C(T)$	0.33	0.66	0.33	0.66

B. Team Scale and Composition

We vary team size while maintaining balanced robot-type composition to study scaling. We group robots into teams of three, with one robot of each type per team. Each LLM agent plans for one team. We evaluate 3, 9, and 15 robots, corresponding to 1, 3, and 5 LLM agents, respectively.

C. Performance and Collaboration Overhead

Task performance is measured by task completion rate (CR) and environment steps (ES). CR is the fraction of task-graph nodes completed by the end of a run, $\text{CR} = |V_{\text{done}}|/|V|$. ES is the number of environment steps used to complete all nodes, up to a maximum.

Collaboration overhead is measured by aggregate agent waiting time (WT), interruption time (IT), and message load (ML). From the LLM agents' MAEIL traces, $\text{WT} = \sum_a \sum_{\hat{t}} \mathbf{1}[q_{a,\hat{t}} = \text{W}] \Delta\hat{t}_{a,\hat{t}}$, $\text{IT} = \sum_a \sum_{\hat{t}} \mathbf{1}[q_{a,\hat{t}} = \text{I}] \Delta\hat{t}_{a,\hat{t}}$, and $\text{ML} = |\hat{M}|$, where a ranges over all LLM agents controlling the robot teams. WT and IT aggregate time spent waiting and handling interruptions, respectively, while ML counts one message per recipient for each send, including the dummy messages sent by `notify`.

V. RESULTS

Agents interact with the environment using the symbolic tools in Tab. II. We use `gpt-5.6-terra` as the underlying LLM for all experiments. We evaluate each setting over three runs per task with $N \in \{3, 9, 15\}$ robots, controlled by one, three, and five LLM agents, respectively. Tab. III reports task performance (CR and ES) and collaboration overhead (WT, IT, and ML). Fig. 7 additionally reports timeout rate (TO), the percentage of runs that terminate before full completion

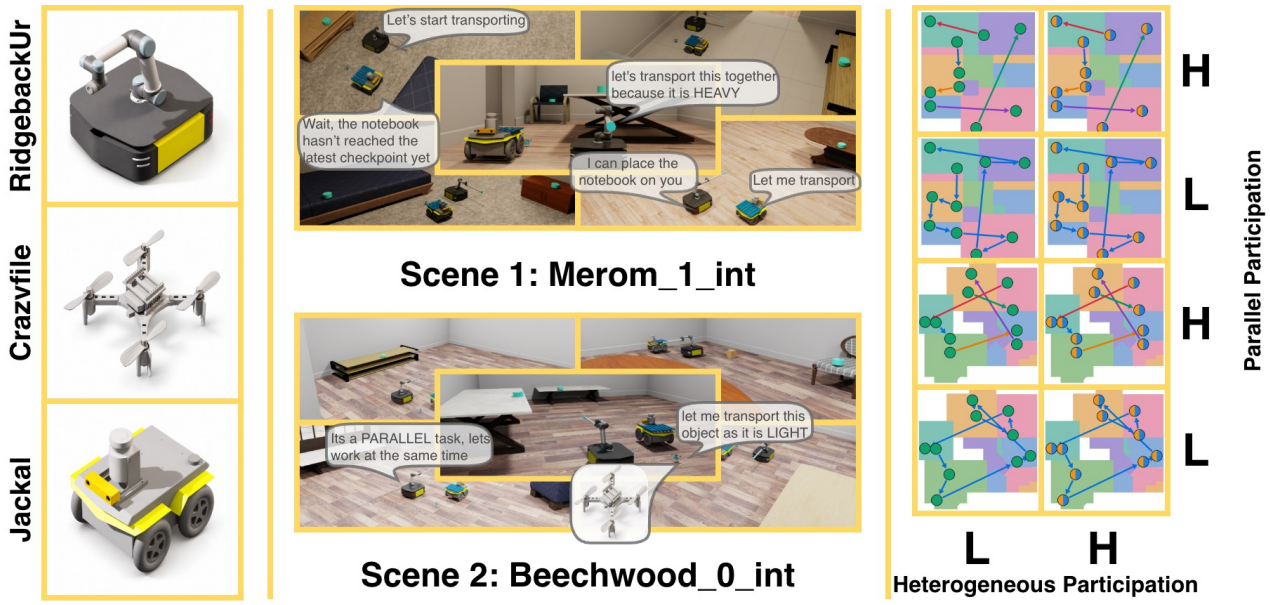


Fig. 5. **Embodied evaluation scenarios and task configurations.** Left: RidgebackUr mobile manipulators, Crazyflie quadrotors, and Jackal mobile robots form balanced teams of $N \in \{3, 9, 15\}$, with equal numbers of each type. Center: scene and task examples from Merom_1_int and Beechwood_0_int. Right: typed evaluation task graphs span low and high $P(T)$ (rows) and $C(T)$ (columns), yielding four configurations per scene and eight tasks in total.

TABLE III

RESULTS BY SCENARIO, TASK CONFIGURATION, AND TEAM SIZE. ENTRIES AVERAGE THREE RUNS IN EACH OF TWO SCENARIOS. METRICS ARE TASK COMPLETION RATE (CR) AND ENVIRONMENT STEPS (ES), MEASURING PERFORMANCE; AND AGENT WAITING TIME (WT), INTERRUPTION TIME (IT), AND MESSAGE LOAD (ML), MEASURING OVERHEAD. ES IS THE EPISODE'S STEP COUNT WHETHER IT COMPLETED OR RAN OUT; WT, IT AND ML ARE UNDEFINED WITH A SINGLE TEAM ($N = 3$) AND THUS OMITTED. SHADING SEPARATES THE FOUR TASK CONFIGURATIONS.

			LL					LH					HL					HH				
	N	Method	CR ↑	ES ↓	WT ↓	IT ↓	ML ↓	CR ↑	ES ↓	WT ↓	IT ↓	ML ↓	CR ↑	ES ↓	WT ↓	IT ↓	ML ↓	CR ↑	ES ↓	WT ↓	IT ↓	ML ↓
S1:Merom_1_int	3	Individual	1.0	2707.0	–	–	–	0.8	5301.7	–	–	–	1.0	2267.7	–	–	–	0.8	5642.7	–	–	–
		Centralized	1.0	1859.0	–	–	–	0.8	5916.3	–	–	–	1.0	1623.0	–	–	–	0.7	5954.0	–	–	–
		Chain	1.0	3326.0	–	–	–	0.9	5228.7	–	–	–	1.0	3021.7	–	–	–	1.0	4561.0	–	–	–
		SHAPE	1.0	3296.3	–	–	–	1.0	5221.7	–	–	–	1.0	1485.3	–	–	–	1.0	4484.0	–	–	–
	9	Individual	1.0	2000.3	38.2	-	–	0.2	5963.7	27.2	–	–	1.0	1553.0	26.0	–	–	1.0	3210.0	36.2	–	–
		Centralized	1.0	3213.7	222.1	97.3	26.7	0.6	4817.7	484.3	211.6	44.0	1.0	1233.3	149.1	93.6	28.0	0.8	2768.3	427.6	309.8	48.0
		Chain	1.0	2007.7	654.9	166.5	28.7	0.3	3353.3	1248.9	308.6	42.0	1.0	796.0	643.6	176.8	20.3	0.8	2230.7	1178.9	313.7	37.3
		SHAPE	1.0	3314.0	125.0	22.3	10.3	1.0	4563.0	231.9	39.5	15.7	1.0	1259.7	117.1	39.5	6.3	1.0	2431.3	114.4	26.8	5.3
	15	Individual	1.0	1973.0	73.9	–	–	0.2	4669.7	87.7	–	–	1.0	1223.7	66.5	–	–	0.7	2196.3	159.6	–	–
		Centralized	1.0	3237.7	441.7	205.7	58.7	0.2	3453.3	1075.1	310.5	74.0	1.0	602.3	139.0	69.6	21.3	0.9	1551.3	771.7	259.1	53.3
		Chain	0.8	1846.3	2255.2	559.5	110.0	0.2	2223.3	2128.6	540.4	109.7	1.0	458.0	1182.2	281.5	57.3	0.6	731.7	2154.9	439.2	72.3
		SHAPE	1.0	3275.0	339.9	95.8	36.3	0.8	5066.3	587.0	122.7	39.3	1.0	1226.0	354.8	103.4	21.3	0.8	2146.3	469.1	153.5	28.7
S2:Beechwood_0_int	3	Individual	1.0	2077.0	–	–	–	0.9	5555.7	–	–	–	0.9	4676.3	–	–	–	0.9	5089.3	–	–	–
		Centralized	0.9	3089.3	–	–	–	0.9	5722.0	–	–	–	1.0	3733.7	–	–	–	0.9	5208.7	–	–	–
		Chain	1.0	1522.3	–	–	–	0.9	4928.0	–	–	–	1.0	2445.7	–	–	–	0.9	5202.3	–	–	–
		SHAPE	1.0	1043.0	–	–	–	0.9	5192.7	–	–	–	1.0	1744.3	–	–	–	0.7	5397.0	–	–	–
	9	Individual	0.9	2616.3	29.6	–	–	0.6	5953.7	36.8	–	–	1.0	1530.3	15.6	–	–	0.9	3479.7	39.9	–	–
		Centralized	1.0	4506.0	216.9	110.0	33.3	0.4	4505.7	544.1	222.8	40.0	1.0	1403.7	87.7	73.3	20.0	1.0	2878.0	560.5	314.3	49.3
		Chain	0.9	2453.7	852.7	240.7	38.3	0.6	3914.0	1108.9	283.4	40.3	0.9	2232.0	828.7	215.8	35.0	0.7	1930.7	1219.5	311.5	37.3
		SHAPE	0.9	3556.7	226.1	68.8	17.7	1.0	4714.3	196.5	48.3	9.7	1.0	2716.3	170.4	40.2	10.3	0.9	3829.3	238.0	54.7	9.7
	15	Individual	1.0	1256.0	44.7	–	–	0.2	4316.0	124.4	–	–	1.0	1438.3	53.3	–	–	0.7	2491.0	56.0	–	–
		Centralized	1.0	4041.3	493.1	222.3	64.0	0.2	3477.0	821.7	250.9	56.0	0.9	4802.7	508.4	273.6	82.7	0.8	2425.3	895.3	401.2	80.0
		Chain	1.0	1551.0	2394.1	574.1	109.3	0.4	1802.3	2250.1	549.7	96.3	1.0	1143.0	1788.3	407.0	79.0	0.7	888.0	2640.0	537.3	97.7
		SHAPE	1.0	1655.3	294.0	120.8	38.3	0.4	5105.0	442.6	91.3	31.7	1.0	1296.0	373.4	92.5	24.0	0.9	2376.7	355.6	127.6	27.7

because they reach either the environment-step limit or the wall-clock time limit. We organize the results around three themes: shared task understanding, agent perspectives, and synchronization costs. We conclude with a shared-channel ablation and a case study of teams grouped by robot type.

A. Shared Task Understanding

An agent's interpretation of task state, including completed goals, ongoing work, and remaining dependencies, directly shapes its future plans. Individual forms this understanding independently. Centralized and Chain exchange task information through messages, while SHAPE maintains an explicit, evolving task state through asynchronous contributions from all agents. Tab. III shows that Individual retains high completion as teams grow when heterogeneous demand is low (LL and HL). With high heterogeneous demand (LH and HH), completion declines as the robot count increases from 3 to 15. SHAPE achieves a higher mean completion rate than Individual at $N = 15$ in both settings with high heterogeneous demand: 0.60 versus 0.20 in LH and 0.85 versus 0.70 in HH. *Shared task understanding becomes especially valuable as teams scale on tasks with high heterogeneous demands.*

B. Agent Perspectives

Different collaboration patterns determine how agents contribute their perspectives to planning. We compare Centralized, which relies on a leader, Chain, which incorporates agents' perspectives sequentially, and SHAPE, which allows agents to asynchronously revise shared tasks. At $N = 15$, Tab. III shows that Chain and SHAPE achieve higher LH completion than Centralized (CR 0.30 and 0.60 versus 0.20), while Centralized and SHAPE achieve higher HH completion than Chain (CR 0.85 versus 0.65). Fig. 6 shows the same advantage across all tasks: all baselines' CR decline with team size, while SHAPE achieves the highest mean CR at $N = 9$ and $N = 15$. These differences show that the value of agent perspectives depends on task structure: multiple perspectives can help resolve task dependencies, while centralized decisions can help allocate work. SHAPE integrates these perspectives through a shared, revisable task graph. *Diverse perspectives become more valuable as teams grow, task dependencies become more complex, and effective plans are harder to identify, but they must be effectively integrated to improve task completion.*

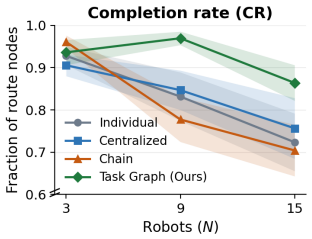


Fig. 6. Mean CR versus team size over all settings (two scenarios, four task configurations, and three runs per setting). Shading indicates standard deviation across runs. Baselines decline with scale; SHAPE peaks at $N = 9$ and remains highest at $N = 15$.

C. Timely Agreement and Synchronization Costs

Synchronization aligns agents' plans but can delay execution as teams grow. Fig. 7 reports waiting time (WT), interruption time (IT), message load (ML), and timeout rate (TO) across team sizes. Individual has low overhead because its LLM agents do not communicate. Chain requires sequential responses, which can delay execution, and has the highest timeout rate. As the robot count increases from 9 to 15, Centralized and Chain incur increasing WT, IT, and ML. At $N = 15$, averaging across task configurations and scenarios, SHAPE's WT, IT, and ML are roughly 37.5%, 54.5%, and 49.5% lower than Centralized's, respectively, and all are lower than Chain's. *SHAPE's shared task state and asynchronous notifications support coordination as teams grow, enabling timely agreement with low collaboration overhead.*

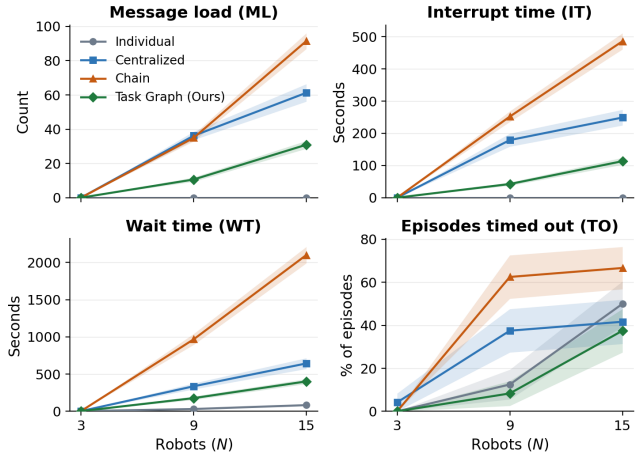


Fig. 7. Mean message load, interruption time, waiting time, and timeout frequency versus team size. Shading indicates standard deviation across runs. WT and IT sum time over LLM agents; ML counts messages per recipient. TO is the percentage of runs terminated before completion by environment-step or wall-clock limits.

D. Shared-Channel Ablation

We ablate SHAPE's structured task graph to isolate its contribution to collaboration. We replace it with an unstructured shared channel that provides full message history while retaining `notify`, referred to as Shared-Channel. In Merom, SHAPE improves completion on low-parallelism, high-heterogeneity tasks at $N = 9$ (1.0 versus 0.7) and reduces environment steps on high-parallelism, low-heterogeneity tasks at both $N = 9$ (1259.7 versus 3515.0) and $N = 15$ (1226.0 versus 1431.0). The methods are otherwise comparable, except on high-parallelism, high-heterogeneity tasks at $N = 15$, where Shared-Channel achieves higher completion (0.9 versus 0.8). *These mixed results highlight the potential of structured task state over full message histories and motivate further study of structured representations for task selection, dependency resolution, and synchronization.*

E. Case study

We study collaboration when robot capabilities are distributed across agents, making cross-agent coordination necessary. We regroup $N = 9$ robots into three teams by robot type, with each team controlled by one LLM agent. We select high $C(T)$ tasks, which require contributions

TABLE IV

CASE STUDY FOR $N = 9$ ROBOTS GROUPED INTO THREE TEAMS BY ROBOT TYPE, WITH ONE LLM AGENT PER TEAM.

Task	Method	CR↑	ES↓	WT↓	IT ↓	ML ↓
LH	Individual	0.17	5744.0	34.2	–	–
	Chain	0.33	5126.7	871.8	273.2	52.7
	Centralized	0.43	4889.0	618.8	231.5	61.3
	SHAPE	0.37	5504.0	408.0	115.9	24.0
HH	Individual	0.57	5716.7	43.5	–	–
	Chain	0.40	5289.7	752.4	270.9	42.7
	Centralized	0.77	4722.3	576.1	243.1	62.7
	SHAPE	0.80	5265.3	521.3	156.6	27.0

from different robot types and thus cross-agent coordination. Tab. IV shows that SHAPE reduces WT, IT, and ML compared with Centralized. Its completion rate is lower in LH (0.37 versus 0.43), but higher in HH (0.80 versus 0.77), where more parallel opportunities exist. *SHAPE reduces collaboration overhead even when task completion requires coordination across agents with different robot capabilities.*

VI. CONCLUSION

We studied how parallel and heterogeneous participation shape scaling in LLM-powered robot teams across eight embodied tasks. SHAPE achieves higher average completion with lower overhead than Centralized and Chain across scales. These findings motivate adaptive multi-robot planning that maintains shared task understanding, integrates agent perspectives, and balances timely agreement against synchronization costs. We identify three directions for future work. First, letting agents request robot types would make team composition part of the collaboration problem. Second, effective methods for encoding task attributes, including capability requirements and dependencies, could reduce synchronization costs. Third, using task structure to retrieve and compose context could support larger teams on long-horizon and distributed tasks.

ACKNOWLEDGMENTS

Research supported by the NVIDIA Academic Grant Program using NVIDIA RTX PRO 6000 Max-Q Workstation Edition GPUs.

REFERENCES

[1] G. A. Cardona and C.-I. Vasile, "Planning for heterogeneous teams of robots with temporal logic, capability, and resource constraints," *The International Journal of Robotics Research*, vol. 43, no. 13, pp. 2089–2111, 2024.

[2] A. Calvo and J. Capitán, "Heterogeneous multirobot task allocation for long-endurance missions in dynamic scenarios," *IEEE Transactions on Robotics*, vol. 41, pp. 6494–6513, 2025.

[3] A. Prorok, M. Malencia, L. Carlone, G. S. Sukhatme, B. M. Sadler, and V. Kumar, "Beyond robustness: A taxonomy of approaches towards resilient multi-robot systems," *arXiv preprint arXiv:2109.12343*, 2021.

[4] Z. Mandi, S. Jain, and S. Song, "Roco: Dialectic multi-robot collaboration with large language models," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 286–299.

[5] S. S. Kannan, V. L. Venkatesh, and B.-C. Min, "Smart-llm: Smart multi-agent robot task planning using large language models," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 12 140–12 147.

[6] K. Liu, Z. Tang, D. Wang, Z. Wang, X. Li, and B. Zhao, "Coherent: Collaboration of heterogeneous multi-robot system with large language models," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 10 208–10 214.

[7] G. Heppner, D. Oberacker, A. Roennau, and R. Dillmann, "Behavior tree capabilities for dynamic multi-robot task allocation with heterogeneous robot teams," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 4826–4833.

[8] J. Chen, H. Yang, Z. Liu, and C. Joe-Wong, "The five ws of multi-agent communication: Who talks to whom, when, what, and why - a survey from MARL to emergent language and LLMs," *Transactions on Machine Learning Research*, 2026, survey Certification. [Online]. Available: https://openreview.net/forum?id=LGsed0QQVq

[9] H. Yang, Y. Mao, Z. Zhang, Y. Yao, J. Chen, T. Lan, and C. Joe-Wong, "DIG-TAG: Representing open-ended collaboration in LLM-based multi-agent systems structurally," Project page, 2026. [Online]. Available: https://happyeureka.github.io/dig-tag/

[10] H. Yang, N. Nourzad, S. Chen, M. Siew, J. Chen, and C. Joe-Wong, "Coop^2: Defining, observing, and repairing cooperation in llm multi-agent systems," 2026. [Online]. Available: https://arxiv.org/abs/2603.00349

[11] H. Sun, S. Zhang, L. Niu, L. Ren, H. Xu, H. Fu, F. Zhao, C. Yuan, and X. Wang, "Collab-overcooked: Benchmarking and evaluating large language models as collaborative agents," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025, pp. 4922–4951.

[12] Y. Kim, K. Gu, C. Park, C. Park, S. Schmidgall, A. A. Heydari, Y. Yan, Z. Zhang, Y. Zhuang, Y. Liu *et al.*, "Towards a science of scaling agent systems," *arXiv preprint arXiv:2512.08296*, 2025.

[13] J. Li, Q. Zhang, Y. Yu, Q. Fu, and D. Ye, "More agents is all you need," *arXiv preprint arXiv:2402.05120*, 2024.

[14] C. Qian, Z. Xie, Y. Wang, W. Liu, K. Zhu, H. Xia, Y. Dang, Z. Du, W. Chen, C. Yang *et al.*, "Scaling large language model-based multi-agent collaboration," in *International Conference on Learning Representations*, vol. 2025, 2025, pp. 41 488–41 505.

[15] Y. Zhang, F. Liu, Y. Shan, X. Huang, X. Yang, Y. Zhu, X. Cheng, C. Liu, K. Zeng, T. J. Zhang *et al.*, "Silo-bench: A scalable environment for evaluating distributed coordination in multi-agent llm systems," in *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026, pp. 29 379–29 398.

[16] Y. Chen, J. Arkin, Y. Zhang, N. Roy, and C. Fan, "Scalable multi-robot collaboration with large language models: Centralized or decentralized systems?" in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 4311–4317.

[17] H. Yang, N. Nourzad, S. Chen, and C. Joe-Wong, "Cube: Collaborative multi-agent block-pushing environment for collective planning with llm agents," in *Workshop on Scaling Environments for Agents*, 2025.

[18] S. Srivastava, C. Li, M. Lingelbach, R. Martín-Martín, F. Xia, K. E. Vainio, Z. Lian, C. Gokmen, S. Buch, K. Liu *et al.*, "Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments," in *Conference on robot learning*. PMLR, 2022, pp. 477–490.

[19] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun *et al.*, "Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 80–93.

[20] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: a survey: S. gronauer, k. diepold," *Artificial intelligence review*, vol. 55, no. 2, pp. 895–943, 2022.

[21] E. Debie, K. Kasmarik, and M. Garratt, "Swarm robotics: A survey from a multi-tasking perspective," *ACM Computing Surveys*, vol. 56, no. 2, pp. 1–38, 2023.

[22] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," *arXiv preprint arXiv:2210.03629*, 2022.

[23] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu *et al.*, "Autogen: Enabling next-gen llm applications via multi-agent conversation," *arXiv preprint arXiv:2308.08155*, 2023.

[24] H. Zhang, W. Du, J. Shan, Q. Zhou, Y. Du, J. B. Tenenbaum, T. Shu, and C. Gan, "Building cooperative embodied agents modularly with large language models," in *International Conference on Learning Representations*, vol. 2024, 2024, pp. 19 373–19 401.

[25] J. Chen, C. Yu, X. Zhou, T. Xu, Y. Mu, M. Hu, W. Shao, Y. Wang, G. Li, and L. Shao, "Emos: Embodiment-aware heterogeneous multi-robot operating system with llm agents," in *International Conference on Learning Representations*, vol. 2025, 2025, pp. 56 670–56 690.

[26] W. Yu, J. Peng, Y. Ying, S. Li, J. Ji, and Y. Zhang, "Mhrc: Closed-loop decentralized multi-heterogeneous robot collaboration with large language models," *arXiv preprint arXiv:2409.16030*, 2024.

[27] H. Yang, J. Chen, M. Siew, T. Lorido-Botran, and C. Joe-Wong, "Llm-powered decentralized generative agents with adaptive hierarchical knowledge graph for cooperative planning," *arXiv preprint arXiv:2502.05453*, 2025.

[28] M. Brenner and B. Nebel, "Continual planning and acting in dynamic multiagent environments," in *Proceedings of the 2006 international symposium on Practical cognitive agents and robots*, 2006, pp. 15–26.

[29] K. Obata, T. Aoki, T. Horii, T. Taniguchi, and T. Nagai, "Lip-llm: Integrating linear programming and dependency graph with large language models for multi-robot task planning," *IEEE Robotics and Automation Letters*, vol. 10, no. 2, pp. 1122–1129, 2024.

[30] T. N. Canh, T. T. Viet, P. Van Dinh, X. HoangVan, and N. Y. Chong, "Osdag: Online scheduling for efficient multi-robot collaboration," *arXiv preprint arXiv:2606.15255*, 2026.

[31] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International journal of robotics research*, vol. 23, no. 9, pp. 939–954, 2004.

[32] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *The International journal of robotics research*, vol. 32, no. 12, pp. 1495–1512, 2013.

[33] X. Kang, S.-W. Lee, H. Liu, Y. Wang, and Y.-L. Kuo, "Moving out: Physically-grounded human-AI collaboration," in *Forty-third International Conference on Machine Learning*, 2026. [Online]. Available: https://openreview.net/forum?id=uIsdcoSD3G

[34] M. Li, S. Zhao, Q. Wang, K. Wang, Y. Zhou, S. Srivastava, C. Gokmen, T. Lee, L. E. Li, R. Zhang *et al.*, "Embodied agent interface: Benchmarking llms for embodied decision making," *Advances in Neural Information Processing Systems*, vol. 37, pp. 100 428–100 534, 2024.