nature communications

Article

https://doi.org/10.1038/s41467-026-72797-8

Proactive collaboration via autonomous interaction

Received: 30 July 2025

Accepted: 17 April 2026

Published online: 09 May 2026

Check for updates

Peiyan Li[1,4], Shuyuan Zhang[1,4], Wenju Yang[2,4], Yitong Hu[2], Yuhang Wang[2], Dunzheng Wang[1], Hang Li[2], Di Guo[2] ✉ & Huaping Liu[1,3] ✉

Human teams excel at dynamically restructuring both task assignments and team composition in response to emerging challenges, proactively recruiting or releasing members as needed. This capacity for autonomous adaptation is a cornerstone of effective teamwork, yet remains difficult to achieve in heterogeneous multi-robot systems, which typically operate under fixed team configurations or adapt only responsively to external disruptions. In this work, we present a systematic investigation of the Proactive Collaboration paradigm for robot teams, where the working team autonomously recruits or releases members as tasks evolve. We implement this paradigm by equipping robots with the developed Autonomous Interaction framework, which utilizes need-driven multi-round communication to facilitate discussions over task progress, negotiated task allocation, and dynamic team resizing. Through real-world and simulated experiments, we demonstrate that our framework effectively realizes the Proactive Collaboration. By resolving capability gaps via anticipatory planning and minimizing action redundancy, it yields consistent and measurable gains in team efficiency and robustness. Our findings suggest that enabling individual-level initiative may offer a promising pathway toward more adaptive and cohesive collective behavior in multi-robot systems.

Real-world scenarios, ranging from indoor services to disaster response and planetary exploration, increasingly require multi-robot teams to address tasks too complex for individual agents[1,2]. While heterogeneous robot teams provide diverse capabilities to handle environmental variability[3–7], the challenge lies in the unpredictable evolution of task demands during execution. Given this unpredictability, it is unrealistic to determine the optimal multi-robot team composition in advance or to deploy an entire large team simultaneously. Instead, the adaptability of the team is critically enhanced by allowing agents to be added or removed as needed, enabling them to autonomously redistribute tasks and solicit collaboration under uncertainty. Therefore, it is necessary for the team to support on-demand team reconfiguration during task execution, a paradigm that imposes exceptionally high demands on the robots' collaboration and communication strategies[8].

For robot teams united by a shared goal, collaboration paradigms are generally categorized based on whether the team composition evolves during task execution: Fixed and Dynamic paradigms[1,9]. The former operates with a pre-determined membership, while the latter allows the composition to be reshaped by adding or removing team members. Fixed Collaboration is widely utilized due to its simple design and ease of deployment, often initiated by manual determination or algorithmic optimization[10–13]. However, this rigid structure inherently results in the team lacking the resilience necessary to adapt to dynamic environmental changes or unexpected failures. Conversely, Dynamic Collaboration significantly enhances the adaptability

[1]Department of Computer Science and Technology, BNRIST, Tsinghua University, Beijing, China. [2]School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China. [3]Institute for Embodied Intelligence and Robotics, Tsinghua University, Beijing, China. [4]These authors contributed equally: Peiyan Li, Shuyuan Zhang, Wenju Yang. ✉e-mail: guodi.gd@gmail.com; hpliu@tsinghua.edu.cn

of the robot team and holds broad potential for resolving complex, evolving tasks. Despite these advantages, realizing such dynamic collaboration faces challenges regarding the underlying mechanisms of adaptation. Specifically, identifying the exact timing and precise requirements for restructuring remains a critical bottleneck that has significantly hindered its widespread application[14].

To elucidate how existing systems tackle this bottleneck, we analyze the triggers driving team composition changes, categorizing dynamic collaboration into two distinct paradigms: Responsive Collaboration and Proactive Collaboration (Fig. 1). As contrasted against the rigid failure of Fixed Collaboration, the key distinction between these two dynamic paradigms lies in whether the team reconfiguration is driven by passive external intervention or active internal foresight.

The Responsive Collaboration paradigm addresses scenarios where team composition changes are imposed by external events rather than the working team's initiative, such as robot failures or the arrival of new agents. This category encompasses research domains such as ad hoc teamwork[15–19] and dynamic team formation[20,21], which typically rely on mechanisms like market-based auctions[20], opportunistic encounters[21], or open-team policy learning[18,22]. However, external entities or algorithms often lack full situational awareness of the working team's specific context, preventing timely and precise interventions. Simultaneously, the working team operates without anticipating these composition changes. Consequently, task reallocation occurs only after the team membership has been externally altered. This passive adaptation limits the team's ability to effectively resolve capability gaps, leading to delayed, misaligned, and inefficient execution trajectories.

In contrast, human teams typically exhibit the Proactive Collaboration paradigm, characterized by the team's autonomous decision to dynamically reshape its composition as the task evolves. Rooted in cognitive empathy and shared situational awareness, experts in high-stakes domains such as emergency medicine and disaster response routinely form temporary coalitions to address urgent challenges[23–25]. In these scenarios, teams do not wait for external dispatchers to identify needs (Responsive). Instead, they actively call for specialists (e.g., Rapid Response Teams) to bridge capability gaps, thereby mitigating delays and information asymmetry[23,26–28]. Recognizing that autonomous adaptation rooted in cognitive empathy is a key driver of collective intelligence[29–31], we aim to realize this mechanism in robotic systems. To support such autonomy, individual agents must possess robust capabilities in situated perception, semantic reasoning, and autonomous arbitration[32]. Moving beyond existing help-seeking frameworks that rely on human-in-the-loop assistance[33–37], we claim that Proactive Collaboration requires agents to autonomously determine when to seek help, whom to ask, and what to communicate. By mastering these dimensions, the team can anticipate capability gaps and proactively recruit specialists or release redundant members, ensuring an aligned and efficient execution flow.

However, realizing such a mechanism remains an open problem due to three complexities. First, determining when to initiate interaction is critical, requiring agents to semantically identify help-seeking needs and the team to optimize the timing for reconfiguration. Second, deciding whom to target involves selecting appropriate helpers from internal members or external candidates. This process complicates task planning, as the system must dynamically determine whether to recruit new members or release redundant ones to align with varying capabilities. Finally, establishing what to communicate necessitates an effective mechanism to synchronize these decisions, ensuring smooth collaboration throughout the process.

To overcome these complexities, we identify Foundation Models as the ideal cognitive foundation, leveraging their emerging capabilities in robotic reasoning, ranging from embodied planning[38,39] and task allocation[40–45] to code generation[46–49], as well as in the social simulation of complex collaborative dynamics[50,51]. Building on this, we propose the Autonomous Interaction framework that integrates individual robot intelligence with a multi-robot coordination mechanism. Each robot follows a two-layer architecture: an Embodiment Layer that closes the perception, action, and reflection loop for situated reasoning and embodied task execution[52,53], and a Communicative Reasoning Layer that converts local states and reflections into structured, goal-directed messages. On top of this per-robot capability, the team coordination is organized by Autonomous Need-Driven Communication, which triggers multi-round dialogs when necessary. This allows robots to deliberate over task progress, negotiate assignments, and dynamically recruit or release teammates to match the evolving requirements.

In summary, this work establishes an Autonomous Interaction framework that realizes Proactive Collaboration by enabling heterogeneous robot teams to anticipate capability gaps and restructure their composition. We further show that Large Language Models are capable of leveraging individual initiative to augment team performance. Finally, we validate the performance across both real-world experiments and extensive simulations, supported by our introduction of DynaTeamTHOR, a benchmark designed for the systematic evaluation of dynamic collaboration.

Results

This section details the implementation and systematic evaluation of the Proactive Collaboration paradigm. We begin by defining the task

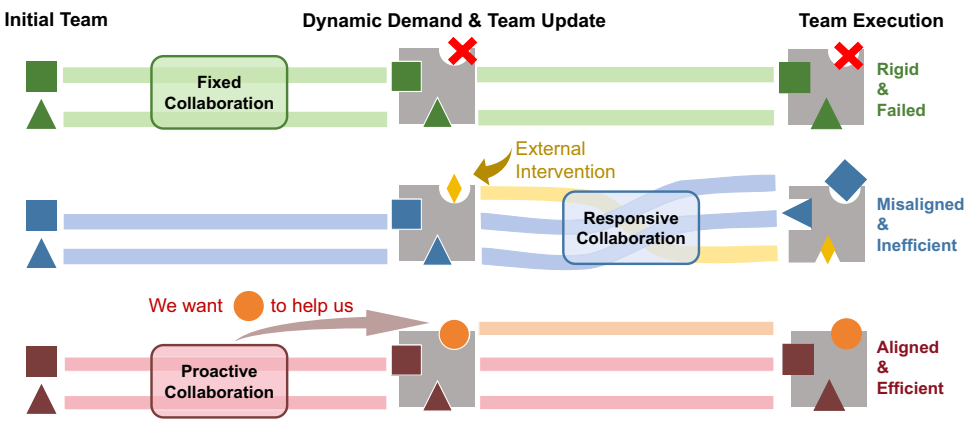


Fig. 1 | Comparison of multi-robot team coordination paradigms. Geometric shapes represent heterogeneous agent capabilities, while slotted blocks denote dynamic execution demands. In Fixed Collaboration, a static team lacks the resilience to resolve capability gaps (red cross), leading to rigid and failed execution. Responsive Collaboration adapts passively after external intervention, resulting in misaligned and inefficient execution. Conversely, Proactive Collaboration autonomously anticipates needs and preemptively recruits required specialists (orange circle), ensuring aligned and efficient execution trajectories.

formulation and experimental design, which establishes the criteria for evaluating fixed, responsive, and proactive team collaboration paradigms (Sec. 2.1). Next, we introduce the Autonomous Interaction framework that integrates individual robot intelligence with a multi-robot coordination mechanism to support efficient teamwork (Sec. 2.2). Building on this architecture, we validate the feasibility and effectiveness of the Proactive Collaboration paradigm through real-world experiments, demonstrating the system's adaptability to physical constraints and changing task requirements (Sec. 2.3). To further substantiate the algorithmic effectiveness of our method, we employ the scalable DynaTeamTHOR simulation benchmark to demonstrate its advantages over baselines under consistent paradigm settings (Sec. 2.3). Finally, we conduct a fine-grained failure analysis to elucidate the mechanisms underlying the observed performance gains and failure mode shifts (Sec. 2.5).

Experimental setup

Embodied Tidying-up Task. We evaluate multi-robot collaboration on an embodied tidying-up task[54,55]. In this setup, a heterogeneous team acts within an indoor environment to explore, identify misplaced objects, infer their correct goal locations, and physically relocate them. Crucially, the task may necessitate coordinated interaction (e.g., clearing obstacles for a teammate) whenever the requirements exceed individual capabilities. Representative real-world task instances are shown in Fig. 2. Overall, this process demands both high-level semantic reasoning and precise team coordination.

Team Model. To implement collaboration under practical constraints, we introduce a team model that separates agents inside the active operation (the working team) from available resources outside the workspace (the robot pool). This structured boundary reflects the resource management required in broader real-world applications, such as buffering standby units in last-mile logistics or staging backup units in warehouse scenarios, as further discussed in Sec. 3. Based on this boundary control, we instantiate three distinct experimental conditions:

- Fixed paradigm (Controlled Baseline): The boundary is closed. The team operates with a fixed membership determined at initialization, prohibiting entry or exit regardless of task demands.
- Responsive paradigm (Baseline): The boundary is semi-permeable but passive. Agents are dispatched from the pool based on the assessment of an external monitoring dispatcher, which determines when intervention is necessary.
- Proactive paradigm (Ours): The boundary is permeable and team-controlled. The working team autonomously manages recruitment or release of members based on evolving task requirements and internal team assessments.

Experimental Design. Our evaluation combines real-world and simulation studies to provide complementary evidence.

In real-world experiments, to validate paradigm feasibility, we deployed the proposed Autonomous Interaction framework as the uniform underlying driver for all three paradigm variants (Fixed, Responsive, and Proactive). This controlled setting serves to demonstrate how the Proactive paradigm unlocks flexibility and efficiency by dynamically adjusting team composition.

Complementing this, simulation experiments focus on verifying algorithmic superiority. Here, we hold the paradigm settings constant and benchmark the proposed Autonomous Interaction against representative baseline methods. Under matched task distributions, we seek to validate that our framework consistently achieves higher efficiency and decision quality. Additionally, this comparative analysis investigates the hypothesis that the efficiency gains of the Proactive paradigm are realized only through a capable interaction mechanism.

Metrics Overview. We focus our evaluation on a unified set of primary metrics characterizing task success and execution efficiency. Specifically, we report Success Rate (SR: a binary indicator of whether an episode fully succeeds) and Partial Success Rate (PS: the fraction of objects correctly placed, simulation only). To measure efficiency, we use Temporal Steps (TS: the number of environment steps until termination) and Total Action Steps (AS: the total number of executed action steps summed across all robots, simulation only). For full definitions and auxiliary metrics, please refer to Sec. 4.3.

Autonomous interaction framework

As illustrated in Fig. 3, the proposed Autonomous Interaction framework integrates individual robot intelligence with a multi-robot coordination mechanism. This hierarchical design bridges high-level semantic reasoning with low-level execution, providing the necessary infrastructure for dynamic role assignment and team reconfiguration (details in Sec. 4).

Individual robot architecture. At the individual level, each robot integrates a dual-layer architecture comprising an Embodiment Layer for physical interaction and a Communicative Reasoning Layer for high-level coordination. (details in Sec. 4.1)

The Embodiment Layer handles the fundamental embodiment cycle: the Visual Semantic Perception module transforms sensory data into structured semantic text (e.g., "pillow on floor, between bed and wall") to facilitate downstream reasoning. Based on this context, the Action Decision module formulates immediate actions executed by the Action Execution module. Crucially, to support proactive need identification, the State Assessment module acts as a continuous self-monitor. It triggers a request for help under two specific conditions: (i) when execution fails (e.g., physical inability), or (ii) when perception identifies a demand exceeding current capabilities (e.g., a "heavy" attribute).

The bridge between individual action and collective coordination is the Communicative Reasoning layer. This layer comprises two distinct modules designed to handle information flow at different stages of the interaction.

- Purposeful Expression: This module facilitates in-communication message generation by ensuring that every utterance is strategically aligned with the robot's current objectives. To ensure coherent and goal-oriented exchange, this module decouples message generation into a two-stage process. During an active dialog, the Thinking Stage first synthesizes the Incoming Message and internal Reflection to establish a clear communicative Purpose. Based on this intent, the Speaking Stage then formulates the Outgoing Message. This explicit abstraction prevents redundant or off-topic utterances, ensuring that communication remains an efficient tool for problem-solving rather than an end in itself.
- Task Summary: This module performs the post-communication task update by anchoring collective consensus into the robot's individual execution pipeline. Once the Autonomous Need-Driven Communication protocol signals the end of a negotiation, this module analyzes the entire Message History to extract actionable consensus. It distillates the multi-robot agreement into a localized task update, effectively translating collective decisions into the robot's next individual Task. By updating the local plan of each participant, this module ensures synchronized role changes and goal alignment across the heterogeneous team without the need for a central orchestrator.

Autonomous need-driven communication. At the collective level, the framework utilizes Autonomous Need-Driven Communication to manage task allocation and member recruitment. This protocol structures multi-robot interactions into sequential, need-driven exchanges based on the evolving task requirements. (details in Sec. 4.2)

Within this protocol, Purposeful Expression functions as the message generation mechanism for each robot. During the

https://doi.org/10.1038/s41467-026-72797-8



Fig. 2 | Four representative tasks in the real-world experiments. a A misplaced tissue should be picked up and returned to its proper place on the desk. **b** A misplaced bottle should be picked up and returned to its proper place in the trash can. **c** A bottle trapped underneath a cart. Pull the cart to free the trapped bottle, then place it in the trash can. **d** A water stain on the floor blocks access to the bottle. Clean the stain, then place the bottle in the trash bin. A video demonstration of the four tasks is provided in Supplementary Video 1. In real-world deployments, the robot team resolves a mixture of these tasks without prior knowledge of their type or number.

communication process, each robot leverages its internal Thinking and Speaking stages to articulate specific needs or offer capabilities, ensuring that every message actively contributes to resolving the current task bottleneck. Through this strategic negotiation, robots navigate complex collaborative scenarios, such as identifying a “capability gap" and proactively recruiting new members from the Robot Pool to join the active team.

https://doi.org/10.1038/s41467-026-72797-8

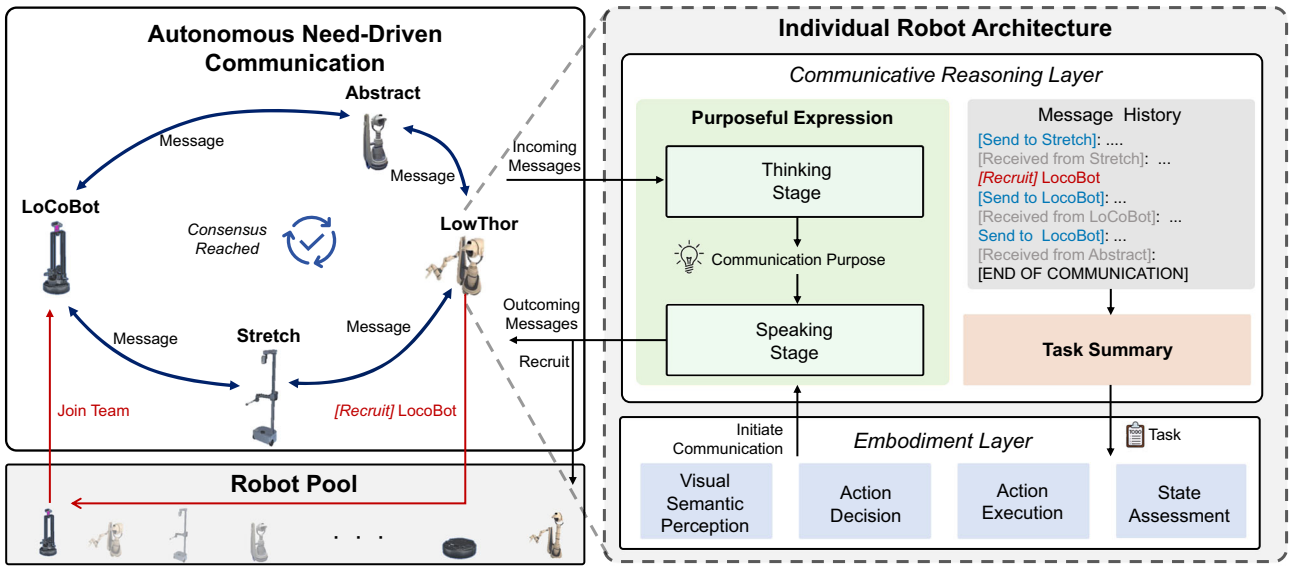


Fig. 3 | Overview of the autonomous interaction framework. Each robot integrates an Embodiment Layer that closes the perception, action, and reflection loop, and a Communicative Reasoning Layer that enables purposeful, goal-directed dialog. Based on reflected capability gaps and task progress, robots autonomously initiate need-driven multi-round communication, recruit or release teammates from the robot pool, and update role assignments via negotiated consensus.

Once the Autonomous Need-Driven Communication concludes that a consensus has been achieved, the Task Summary module performs the post-communication task update by conducting a final consolidation of the entire Message History. It distills the negotiated agreements into updated local plans for all involved agents, ensuring synchronized role transitions and seamless task handovers across the heterogeneous team without a central controller.

Real-world validation

To validate the feasibility of the Proactive Collaboration paradigm in a physical setting, we deployed a heterogeneous team in an office environment comprising four interconnected rooms with both spatial and functional complexity.

Platform and environment. We utilized a pool of five heterogeneous robots positioned outside the main workspace to represent available resources, as shown in Fig. 4a–f. The specific platforms included: (a) Car (AgileX LIMO) and (e) Go2 (Unitree Go2) for rapid perception and navigation; (b) AlienGo (Unitree AlienGo) for locomotion and manipulation; (c) Stretch (Hello Robot Stretch) for complex physical interaction and high-reach manipulation; and (d) Cleaner (iRobot Create3) for surface cleaning. These platforms provide complementary mobility and interaction capabilities. To join the working team, robots from the pool must physically enter the active workspace through a designated entrance, which restricts local perception to deployed robots and formally marks their state transition from an external resource to an active team member. Details of the robotic platforms and sensor configurations are provided in Supplementary Sec. 1.1.1.

Evaluation protocol. We evaluated the three paradigms (Fixed, Responsive, and Proactive) across a total of 60 episodes, following a full factorial design of 3 paradigms × 10 scenarios × 2 initial team configurations. To test adaptability, we designed ten cluttered multi-room scenarios varying in difficulty: 4 instances requiring basic coordination and 6 instances demanding complex interaction (e.g., cart pulling) or auxiliary cleaning.

To emulate the scenario of dispatching a robot team into a room with unknown task requirements, we initialized the teams without prior knowledge of the specific obstacles or messes they would encounter. Consequently, these initial teams may possess varying degrees of capability gaps relative to the actual demand. We modeled this by initiating tasks with two distinct configurations: a basic configuration capable of standard tasks but vulnerable to high-load demands, and an intermediate configuration with improved versatility but remaining incomplete for comprehensive scenarios. This design specifically stresses the system's ability to recognize capability gaps and adjust team composition dynamically under unknown task distributions.

End-to-end performance. As shown in Fig. 4g, the choice of collaboration paradigm significantly impacts the team's overall success and efficiency. The Fixed paradigm struggled with capability gaps, achieving the lowest Success Rate (SR) of 0.45 and the highest average completion time of 931 seconds. The Responsive Paradigm provided a slight improvement (SR 0.55, 894s) by allowing external help, but its performance remained capped by both the dispatch delay and the post-arrival integration latency. In contrast, the Proactive Paradigm achieved a dominant SR of 0.90 with the lowest average time of 735s. This substantial leap validates that enabling the team to autonomously anticipate needs and recruit members significantly reduces failure rates and execution time.

Coordination quality and team dynamics. To decouple coordination stability from resource availability, we analyze the Conditional Success Rate (Fig. 4h). This metric calculates the success rate exclusively for the subset of episodes where a capable team was successfully formed, thereby isolating the impact of the collaboration strategy on execution consistency:

- Inherent Framework Competence (Fixed): For the Fixed paradigm, we calculate this metric over the subset of episodes where the initial team configuration was already theoretically sufficient. The high score of 0.82 serves as a controlled baseline, demonstrating that our underlying Autonomous Interaction framework effectively organizes the team when resources are present and secured from the start.
- Passive Integration Cost (Responsive): In the Responsive paradigm, we isolate the specific episodes where external intervention successfully provided the necessary agents. Despite eventually acquiring the required resources, the paradigm shows a significant drop in conditional success to 0.61. This indicates that the passive introduction of new members often creates coordination misalignment rather than cohesion. Such sudden additions disrupt the existing collaboration, often causing the task to fail even when the team gains the necessary capabilities.

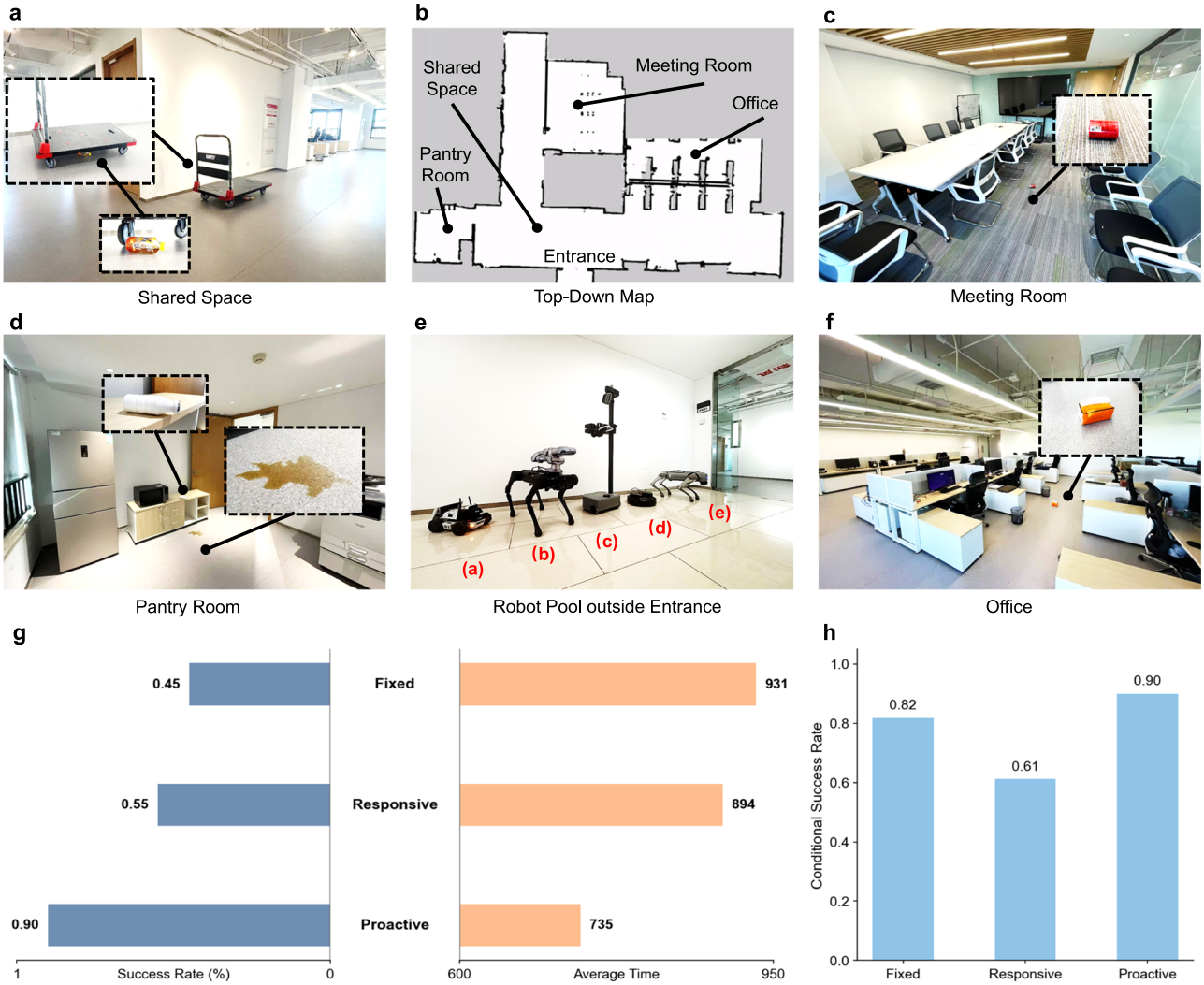


Fig. 4 | Real-world experimental setup and quantitative results for collaborative tidying tasks. a Shared workspace containing a trapped bottle. **b** Top-down map of the overall environment layout. **c** Meeting room with a misplaced bottle. **d** Pantry room with both a water stain and a bottle requiring cleanup. **e** Robot pool located outside the environment entrance containing five robots: (**a**) AgileX LIMO (Car), (**b**) Unitree AlienGo, (**c**) Hello Robot Stretch, (**d**) iRobot Create3 (Cleaner), and (**e**) Unitree Go2. **f** Office space with a misplaced tissue. Target objects in (**a**, **c**, **d**, **f**) appear randomly; these images capture representative instances when tidying is required. **g** Overall Success Rate (%) and Average Time (s) across the Fixed, Responsive, and Proactive paradigms. **h** Conditional Success Rate. This metric measures the success rate calculated exclusively from the subset of episodes where a team with sufficient capabilities was successfully formed (i.e., excluding failures caused purely by the inability to recruit the necessary robot types).

- Proactive Efficiency Gain (Proactive): For the Proactive paradigm, we evaluate episodes where the working team successfully identified and recruited the required members. This approach achieves the highest conditional success rate of 0.90, surpassing even the stable Fixed baseline. This suggests that proactive recruitment acts as more than just a resource buffer. By seeking help at the appropriate moment, the team integrates new members seamlessly before conflicts escalate, turning potential coordination friction into an efficiency gain.

Collectively, these findings demonstrate that the anticipatory planning of reconfiguration is as critical as the resources themselves: whereas passive intervention incurs a coordination penalty, proactive anticipation enables the team to effectively integrate new members to enhance, rather than merely repair, the workflow.

Representative episode. The efficiency of the Proactive paradigm stems from its ability to bridge capability gaps in real-time. As illustrated in Fig. 5 (and representative tasks in Fig. 2), when an initial team encounters a “trapped bottle" requiring heavy manipulation, the system identifies the deficiency and autonomously recruits Stretch from the robot pool. Once Stretch clears the obstacle (the cart), it allows the original team to proceed. Furthermore, the system dynamically releases robots (e.g., Car and Go2) once their tasks are complete, optimizing team size without human intervention. A full video demonstration of this task episode is available in Supplementary Video 2.

Mechanisms of Adaptability. The significant performance advantage of the Proactive paradigm (90% SR vs. 55% in Responsive) stems from two primary mechanisms. First, the team autonomously recruits members during execution, ensuring the group composition aligns with task demands and eliminating the significant waiting latency caused by late external intervention. Second, and crucially, the Proactive paradigm incorporates anticipatory planning to coordinate the team’s state for the specific helper’s arrival. As visualized in Fig. 6, the Responsive paradigm suffers from a prolonged “misaligned action" phase (Integration Latency) even after the required robot eventually arrives. In contrast, the Proactive paradigm orchestrates the recruitment process, enabling direct coordination where the helper integrates into the workflow immediately.

https://doi.org/10.1038/s41467-026-72797-8

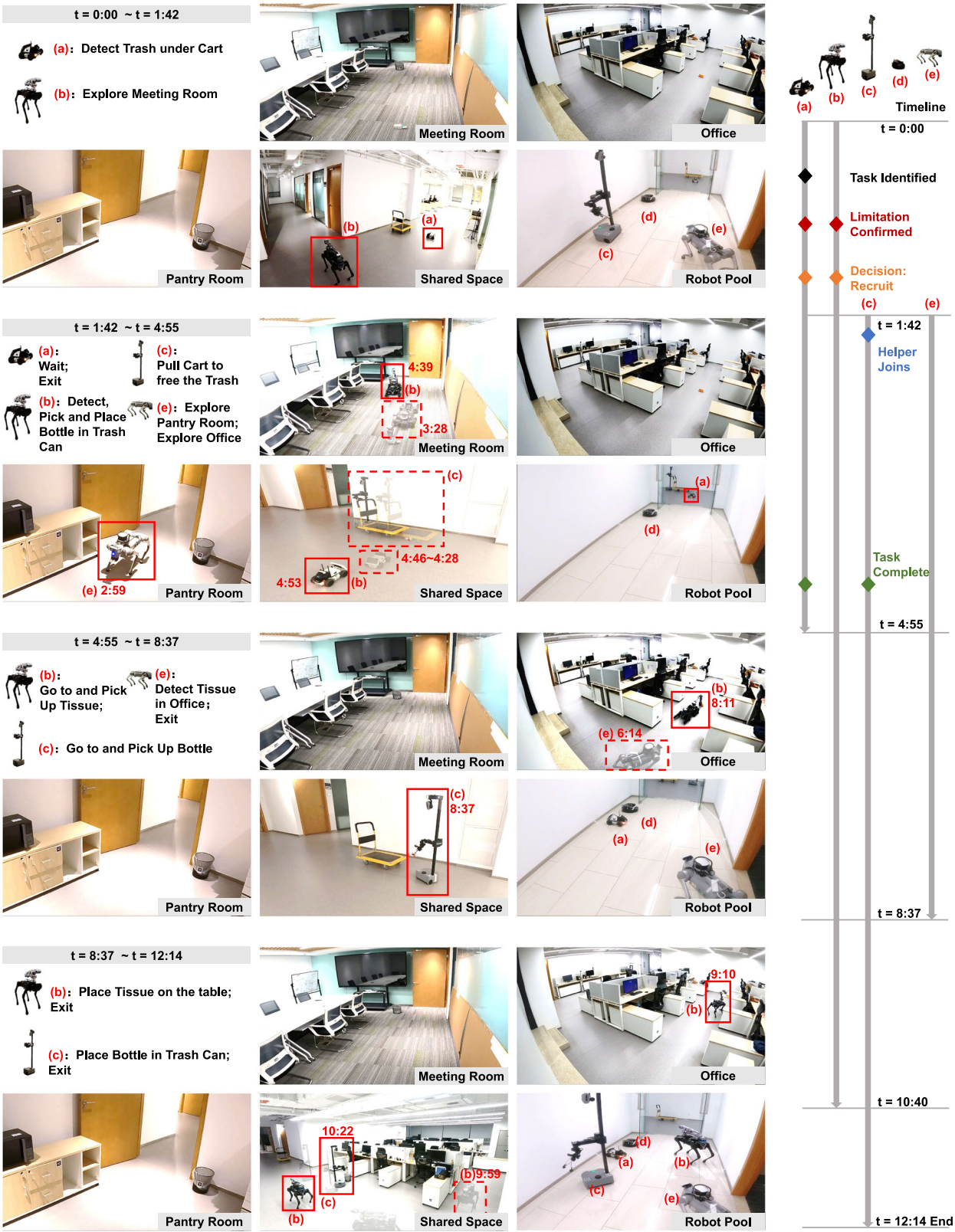


Fig. 5 | Dynamic team composition throughout a single task episode. The episode begins with a team composed of the *Car* and *AlienGo* robots. Upon detecting trash beneath a cart (**Task Identified**), the team initiates a discussion to validate the constraint (**Limitation Confirmed**) and proactively makes a **Decision: Recruit** for *Stretch* and *Go2*. Following the **Helper Joins** event, the *Car* and *Go2* complete their exploration and autonomously exit. Once all misplaced objects are tidied, *Stretch* and *AlienGo* also leave the scene, marking the **Task Complete**. This sequence illustrates the Proactive paradigm's capacity for dynamic, task-driven team reconfiguration.

https://doi.org/10.1038/s41467-026-72797-8

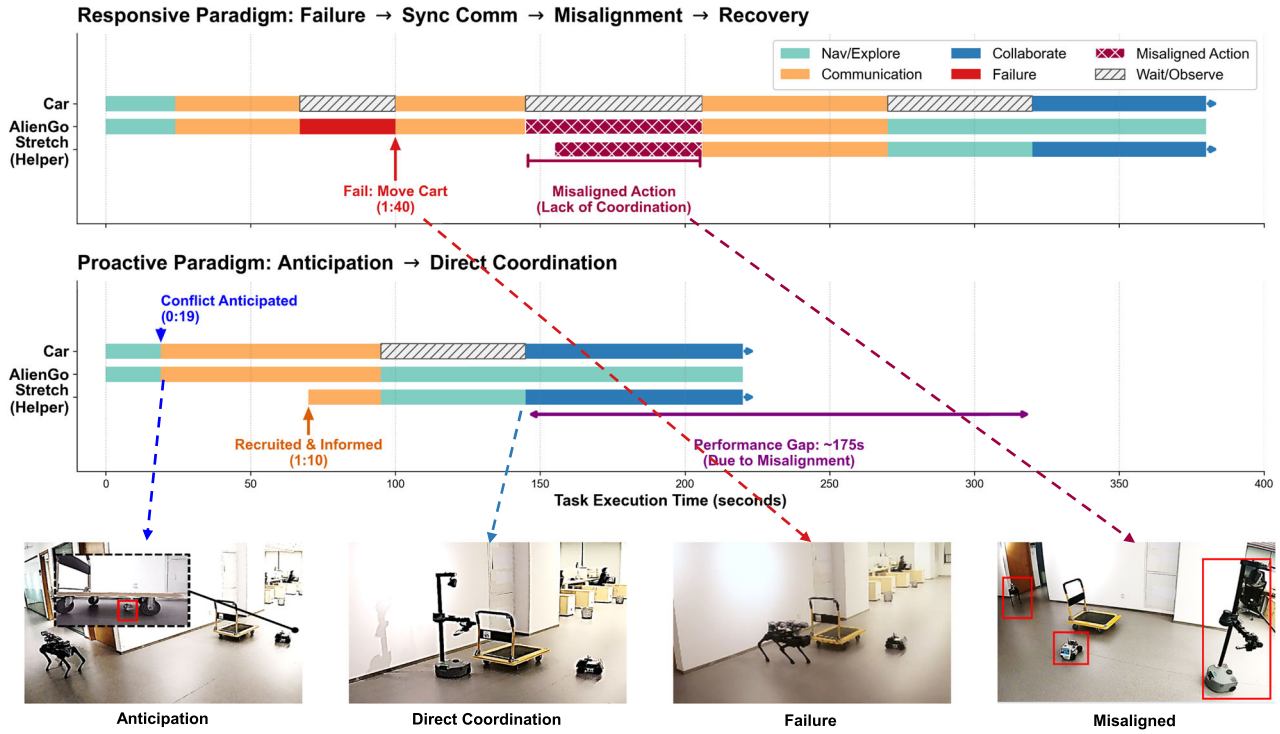


Fig. 6 | Temporal and visual analysis of anticipatory planning. This figure demonstrates the clear advantages of a proactive approach over a responsive one. **Top: Comparison Gantt Charts**. The Responsive paradigm (upper) is marked by a Failure event ($t = 1{:}40$), where the robot physically attempts and fails to move the cart, triggering a Misaligned Action phase where the recruited helper is active but not yet synchronized with the working team. This combination of waiting time and misaligned coordination results in a visible Performance Gap (175s) compared to the proactive case. In contrast, the Proactive paradigm (lower) begins with Conflict Anticipated ($t = 0{:}19$), where the constraint is identified via visual reasoning, allowing the helper to be Recruited & Informed ($t = 1{:}10$) and arrive prepared for the task. **Bottom: Execution Snapshots**. Dashed arrows link the timeline events to key execution snapshots. (Anticipation) The robot visually detects the geometric constraint (a misplaced bottle is trapped under the cart) and initiates planning. (Direct Coordination) The helper (Stretch) navigates directly to the task site, achieving a seamless transition. (Failure) The robot is physically blocked while attempting the infeasible task. (Misaligned) A state of misaligned coordination where the helper (right) operates in an irrelevant area.

Controlled experiments in simulation

Simulation Benchmark: DynaTeamTHOR. To enable statistically grounded comparisons for our proposed Autonomous Interaction framework, we introduce DynaTeamTHOR, which extends the capabilities of the AI2-THOR series[56,57]. The environment features realistic 3D indoor scenes rich in semantic diversity, ranging from living areas to kitchens, each populated with interactive objects and varied furniture layouts. It operationalizes our team model through two key features: (i) A Coordinated Interaction Mechanism that mandates synchronized joint actions to trigger specific state changes (e.g., displacing heavy furniture). This explicitly enables "Trapped Object" scenarios, where target items are occluded by obstacles that require collaborative displacement to access. And (ii) A Dynamic Robot Pool Interface that enables multiple robots to operate simultaneously within the same multi-room workspace, allowing real-time agent instantiation from distinct robot configurations (defined by navigation speed, manipulation range, etc.). This benchmark comprises 10 distinct multi-room layouts (Fig. 7a) and 720 task scenarios, providing a basis for evaluating generalizability across varying spatial structures.

Baselines. To rigorously benchmark algorithmic effectiveness, we compare our framework against representative baselines spanning decision-making and communication strategies. Since a few existing methods naturally support the dynamic team composition required for Proactive Collaboration, we selected and adapted representative strategies from standard heterogeneous robot collaboration:

Regarding planning, we compare against: (i) Random (Baseline 1), serving as a lower bound where agents select skills without deliberation; (ii) Direct_LLM (Baseline 2)[19], representing direct LLM generation with rigid information exchange; and (iii) EMOS (Baseline 3)[40], a recent centralized framework designed for heterogeneous robot coordination, which we adapted for decentralized execution. Crucially, for the Proactive paradigm, all these planning baselines were augmented with our member adjustment mechanism to ensure a fair comparison of dynamic capabilities.

Regarding communication, we evaluate our need-driven protocol against standard sequential mechanisms: Round-Robin Broadcast[58] and Round-Robin Targeted[59]. While both follow a fixed turn-taking order, the former broadcasts to the entire team while the latter addresses specific recipients.

Further evaluations, including comparisons with other adapted frameworks and comprehensive ablation studies, are detailed in Supplementary Sec. 4.

Comparative performance analysis. As illustrated in Fig. 7b, our proposed Autonomous Interaction framework demonstrates superiority across all paradigm variants. In both Fixed and Responsive settings, our method effectively utilizes limited resources to achieve Success Rates (SR) of 48.0% and 56.1%, respectively, significantly outperforming the best baseline scores of 28.8% and 36.9%. The performance gap becomes most pronounced in the Proactive paradigm, where our method achieves a dominant 70.3% SR compared to 37.9% for the best baseline. Furthermore, our approach substantially reduces action overhead (#AS $\approx$ 2335 vs. $>$ 5100 for Baseline 3), proving that anticipatory reasoning enables precise recruitment without redundant actions.

Analysis of proactive efficiency. To investigate the mechanisms driving these gains, we analyzed the relative improvement when shifting from the Responsive to the Proactive paradigm (Fig. 7c). The results reveal that simply enabling proactive recruitment is insufficient

https://doi.org/10.1038/s41467-026-72797-8

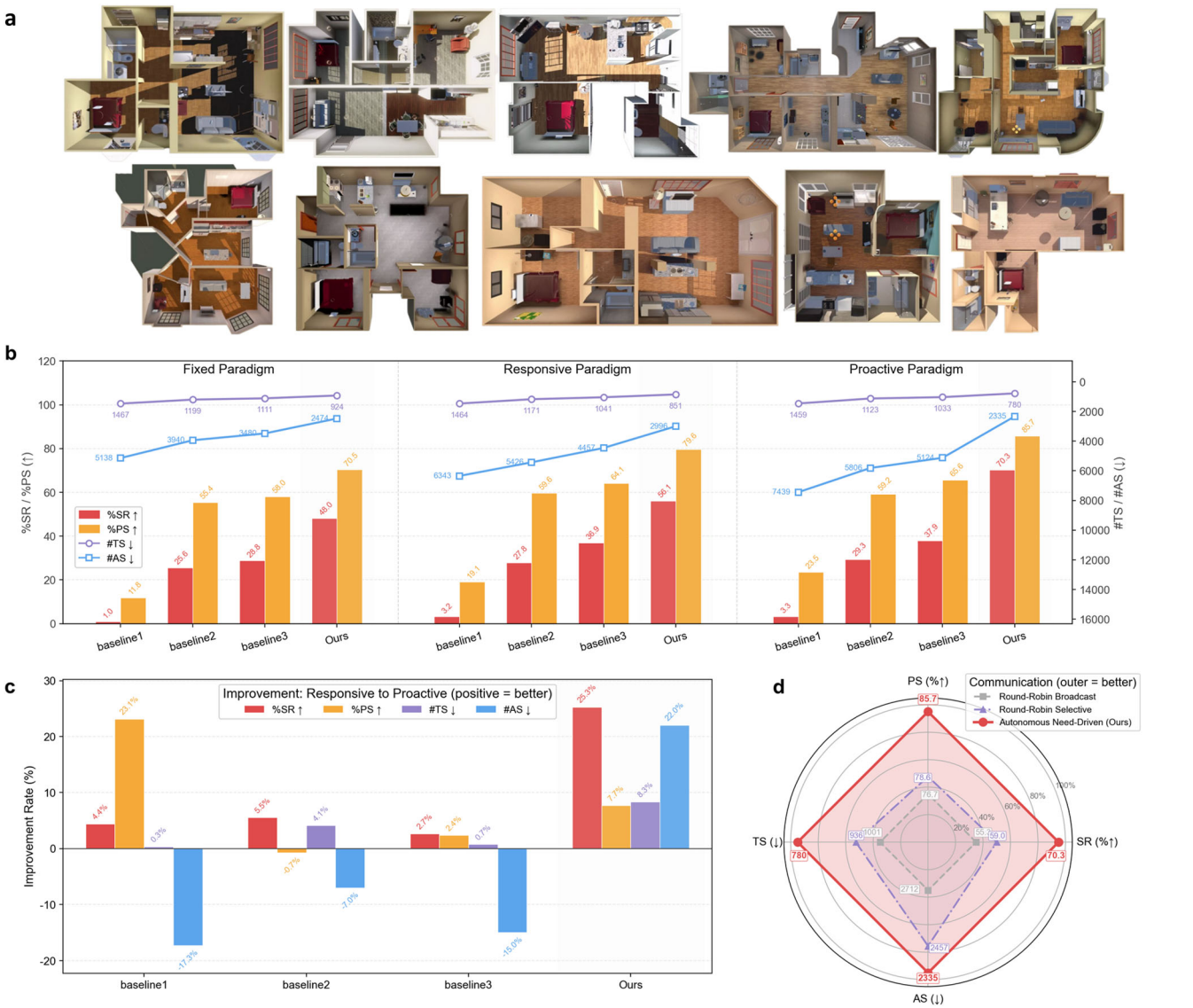


Fig. 7 | Simulation environments and comprehensive performance evaluation. **a** Scenes in the benchmark dataset. Each image shows a top-down view of a distinct house developed in our simulation environment. These feature diverse spatial layouts and object distributions to support systematic evaluation of multi-robot collaboration. **b** Performance comparison of four methods across the Fixed, Responsive, and Proactive paradigms. Bars represent Success Rate (%SR) and Partial Success Rate (%PS) on the left *y*-axis. Lines represent Time Steps (#TS) and Action Steps (#AS) on the right *y*-axis, which is inverted so that higher positions consistently indicate better performance. **c** Improvement rate transitioning from the Responsive to the Proactive paradigm. Values are computed as percentage changes, with signs aligned so that positive values consistently represent improved performance. **d** Radar chart comparing communication strategies. Metrics are normalized and oriented so that positions closer to the outer edge represent better performance.

if the underlying individual intelligence and coordination mechanisms are deficient.

- Inefficiency of Baselines: For Baselines 1, 2, and 3, the shift to the Proactive paradigm yields negligible gains in success rate (< 5% improvement). Critically, their efficiency significantly degrades, as indicated by negative improvement rates in Total Action Steps (e.g., Baseline 1 drops by 17.3%). This suggests that without effective coordination, proactively recruiting more robots leads to resource waste and chaotic execution rather than constructive collaboration.
- Efficiency of Ours: In contrast, our method effectively leverages the proactive freedom, achieving a 25.3% boost in SR alongside a 22.0% improvement in Action Steps. This demonstrates that our method converts proactive freedom into practical efficiency, optimizing the team to solve complex tasks with minimal wasted action.

Impact of communication strategy. Finally, regarding coordination efficiency, we validated the rationality of our interaction mechanism against the sequential communication baselines defined above. As visualized in the radar chart (Fig. 7d), these baseline strategies suffer from redundancy and latency, achieving lower success rates (55-59%). Our approach, which allows flexible, need-based entry into dialogs, encompasses the baselines across all metrics, confirming that flexible communication is essential for efficient multi-robot coordination.

Representative episode. We examine a specific simulation instance to demonstrate the transition from internal coordination to team expansion. When a robot encounters a blocking condition while retrieving an object, it first initiates a help request to its existing teammate. The teammate, however, evaluates its current workload (transporting a saltshaker) and declines the request to maintain individual efficiency. Following this negotiation, the team confirms the lack of internal solutions and autonomously escalates its strategy,

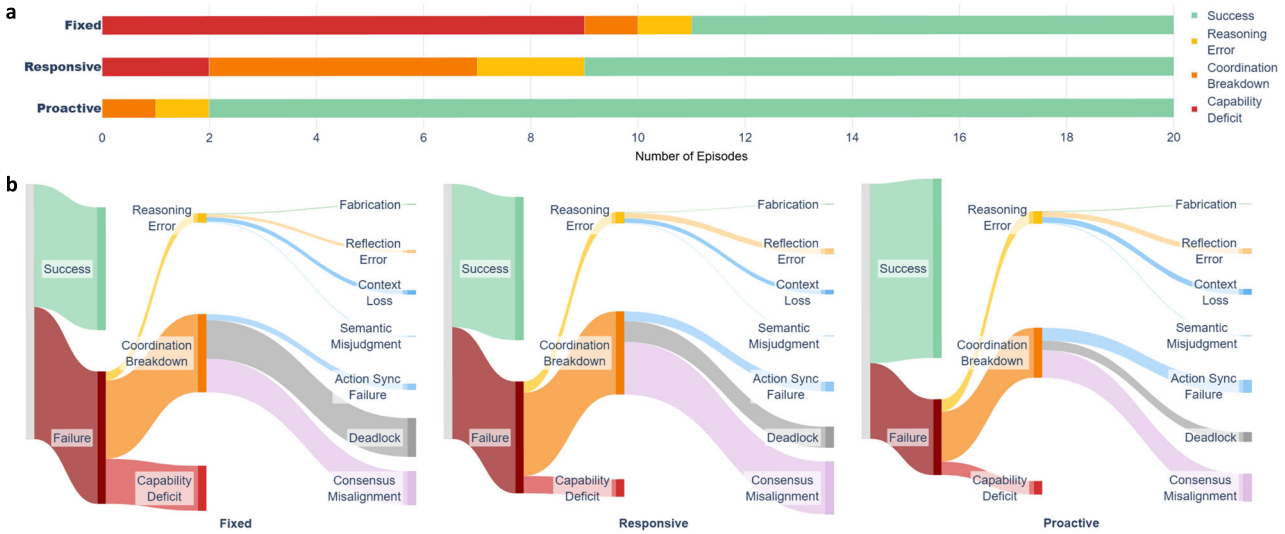


Fig. 8 | Failure analysis across Fixed, Responsive, and Proactive collaboration paradigms. a Real-world failure composition shows a structural shift from capability deficits to coordination breakdowns, which are ultimately resolved in the Proactive paradigm. **b** Fine-grained decomposition in simulation corroborates this trend and further reveals the specific mechanism of coordination failures (e.g., Deadlock). Detailed failure taxonomy and quantitative distribution are provided in Supplementary Sec. 4.5.

dynamically recruiting an available robot from the pool to resolve the conflict. This progression illustrates how team composition evolves during execution to address insufficient internal resources. A full video demonstration of this task episode is available in Supplementary Video 3.

Failure analysis

To elucidate the mechanisms underlying the observed performance differences from a collaboration perspective, we conducted a fine-grained failure analysis. We categorize failures into three distinct mechanistic levels: (i) Capability Deficit (Structural Failure), where the working team fundamentally lacks the required hardware configuration to resolve a task (e.g., inherent limitations of the initial team or failure to recruit necessary capabilities before timeout); (ii) Coordination Breakdown (Inter-Agent Dynamic Failure), where the necessary physical capabilities exist, but execution fails due to flawed multi-agent interaction (e.g., persistent execution deadlocks, unresolved role conflicts, or consensus misalignment during task allocation); and (iii) Reasoning Error (Intra-Agent Cognitive Failure), where the dominant failure is attributable to an individual agent acting in isolation based on flawed LLM logic, independent of team dynamics (e.g., context loss over long sequences, semantic misjudgments, or the fabrication of ungrounded physical properties).

Shift in failure modes. As illustrated in Fig. 8a, the dominant failure sources shift significantly as team adaptability increases. The Fixed paradigm is dominated by Capability Deficit, reflecting the structural rigidity inherent in closed team boundaries. While the Responsive paradigm resolves these structural gaps, it results in a significant increase in Coordination Breakdown. This confirms that reactive recruitment introduces a high interaction penalty, where inopportune arrivals and uncoordinated motion frequently lead to plan misalignment. In contrast, the Proactive paradigm effectively mitigates both structural and dynamic failures. By initiating recruitment and role reassignment early, it minimizes coordination overhead, leaving Reasoning Error (inherent to individual agents) and minor Coordination Breakdown as the residual failure modes.

Fine-grained decomposition in simulation. Large-scale simulation results (Fig. 8b) corroborate this structural failure mode shift and provide deeper insight into the Coordination Breakdown category, particularly regarding Deadlock situations where agents cycle through infeasible actions due to conflicts. The data reveal that Deadlock accounts for ~8.19% of all tasks in the Responsive paradigm, indicating that passive adaptation often traps agents in unseen synchronization dilemmas. However, under the Proactive paradigm, Deadlock frequency decreases to 3.75%. This reduction demonstrates that anticipatory role negotiation acts as a preventative filter, resolving potential conflicts before they manifest in execution and ensuring smoother collaboration flows.

Discussion

This study introduces a Proactive Collaboration paradigm designed to overcome the structural rigidity of fixed robot teams and the operational misalignment inherent in responsive approaches. By equipping agents with the capacity to anticipate task requirements and autonomously adjust team composition, we observed consistent gains in efficiency across simulations and demonstrated feasibility in real-world experiments. These findings indicate the potential for individual-level initiative to cultivate more adaptive and resilient collective behavior in embodied systems.

Emergent behaviors in coordination. To understand the mechanisms driving these aggregate gains, we analyzed the interaction histories from our large-scale simulations (Fig. 9). Notably, distinct social coordination patterns emerged directly from inter-robot communication and decentralized decision-making. We categorize these dynamics into four behaviors analogous to human social coordination:

- Arbitration. In human society, arbitration involves a neutral third party resolving disputes via binding decisions[60,61]. In our setting, Arbitration occurs when an uninvolved robot spontaneously intervenes to resolve a conflict. As shown in Fig. 9, when Stretch and LoCoBot simultaneously attempt a “clock placement" task, a third agent, Cleaner, intervenes by suggesting LoCoBot proceed due to proximity. The conflicting parties accept this unsolicited peer-mediation, illustrating effective conflict resolution without a central coordinator.
- Resource Competition. Analogous to groups contending for limited resources[62,63], we observed Resource Competition where

https://doi.org/10.1038/s41467-026-72797-8

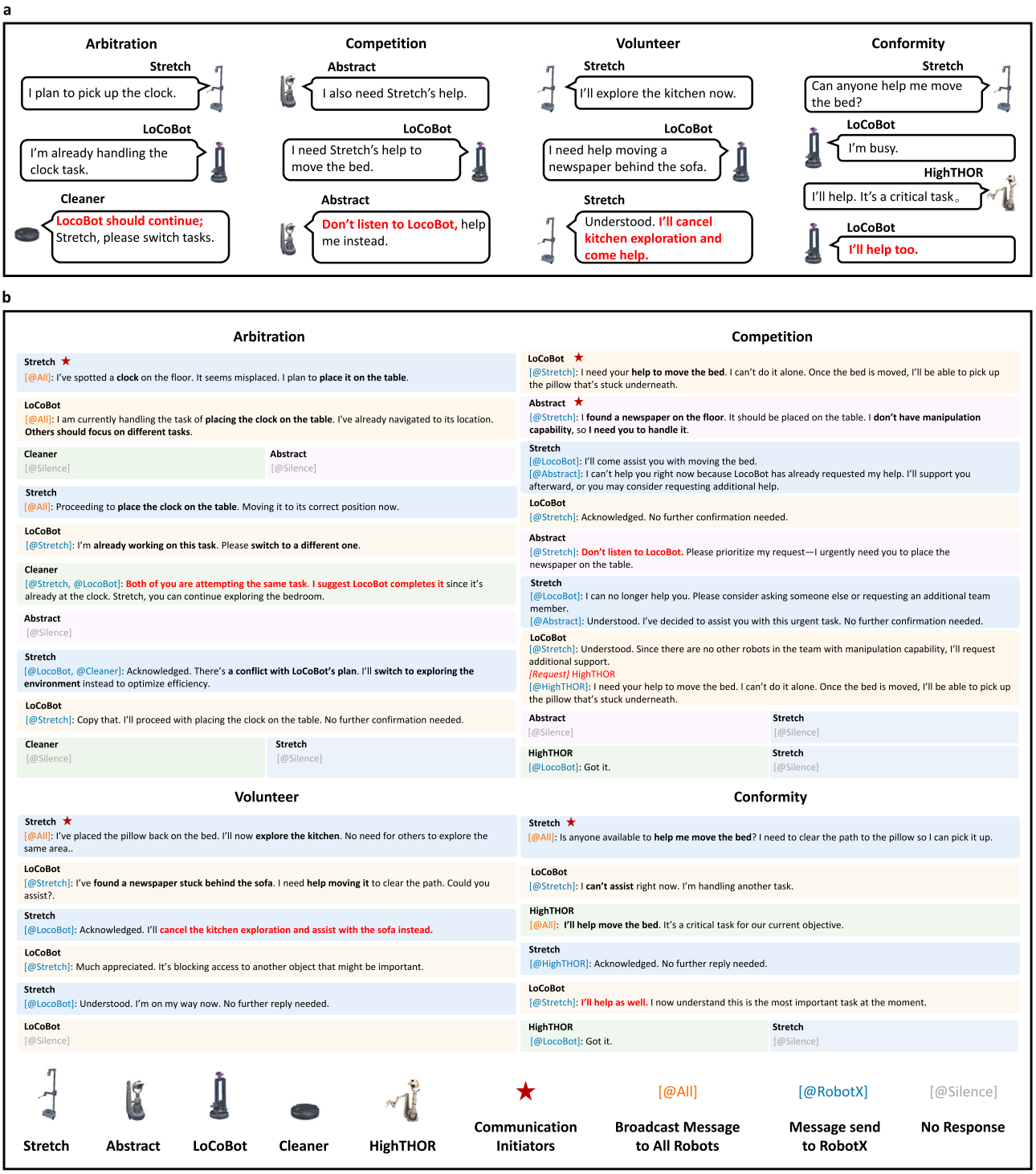


Fig. 9 | Examples of four types of emergent behavior observed in our experiments. a Brief examples of competition, conformity, volunteering, and arbitration. **b** Detailed contents of multi-round dialogs. Arbitration: Robots resolve action conflicts through peer intervention, spontaneously regulating task assignments. Competition: Robots compete for assistance from the same teammate, leading to dynamic reallocation of support. Volunteer: A robot suspends its own plan to assist others, prioritizing team needs over individual goals. Conformity: A robot changes its decision based on peer reasoning and shared team objectives.

multiple agents contend for the same assistance capability. For instance, when both LoCoBot and Abstract request help from Stretch, the initial first-come-first-served logic is overridden when Abstract persuades Stretch by emphasizing task urgency. Consequently, LoCoBot recognizes the unavailability and autonomously recruits HighThor from the robot pool. This illustrates how allocation shifts dynamically under competitive pressure to optimize global efficiency.

- Volunteering. Resembling human volunteerism actions intended to benefit others without obligation[64,65], we define robotic Volunteering as cases where an agent suspends its own plan to assist a teammate without an explicit request. For example, Stretch cancels its planned kitchen exploration upon observing LoCoBot struggling with a "newspaper behind sofa" task. This reflects an emergent form of behavioral empathy, where agents

adapt into a supportive partner or assistant role based on inferred needs rather than rigid role adherence.

- Conformity. Defined as adjusting behavior to align with group norms or goals[66,67], Conformity emerges when robots revise decisions after peer input highlights shared priorities. In our experiments, LoCoBot initially declines a request due to individual commitments but reverses this decision after HighThor emphasizes the task's critical relevance to the overall objective. This adaptation arises through peer reasoning rather than enforcement, ensuring collective success.

The emergence of these sophisticated behaviors offers significant implications for the design of multi-robot systems. The empirical results indicate that decentralized natural language coordination can support timely role negotiation and dynamic subgroup formation during embodied task execution. More broadly, these observed behaviors point to a pathway where local autonomy gives rise to coherent group-level intelligence. This aligns with theories of collective intelligence in human teams, where performance is driven by adaptive interaction patterns and self-organized restructuring[29,31]. However, we acknowledge that the identified taxonomy is likely not exhaustive. We envision that more systematic observation of long-term interactions will reveal additional emergent behaviors, representing a promising direction for multi-robot sociology.

Rationale for the robot pool. The robot pool is grounded in the necessity of efficient resource management within practical deployments. Assigning an entire fleet to a single task consumes excessive resources and is often infeasible due to physical overcrowding and communication bandwidth limits. The pool, therefore, acts as a physical buffer of available resources, such as standby agents waiting at logistics distribution hubs. Conceptually, the pool also represents broader available system capacity. In warehousing scenarios, specialized units remain in staging areas and function as a reserve until primary robots explicitly request assistance. Furthermore, even when robots are engaged in other routine duties, an urgent request can reassign agents from lower-priority tasks. This dynamic effectively treats the entire system as a resource pool relative to the active operation. Consequently, our framework naturally aligns with these realistic constraints.

Limitations and future work. The present study validates the Proactive Collaboration paradigm within indoor collaborative scenarios. While our simulations establish statistical robustness, the real-world experiments demonstrate the feasibility of autonomous role negotiation using representative heterogeneous teams. We acknowledge that further testing across larger sample sizes is required to prove statistical robustness in physical settings. Furthermore, deployment in unstructured or adversarial environments introduces additional complexity regarding communication stability and spatial interference.

Looking forward, we aim to extend the Proactive Collaboration paradigm to more unstructured and constrained environments. Domains such as warehouse automation, disaster search and rescue[1], and subterranean exploration[5] represent promising directions where dynamic team resizing is critical, but communication bandwidth is limited. Future work will explore how this framework for decentralized coordination generalizes to larger and more diverse robot fleets. Furthermore, future research will investigate the integration of more sophisticated empathy-inspired mechanisms to foster flexible and effective proactivity.

Methods

Individual robot architecture

Each robot integrates a two-layer architecture (Fig. 3): an Embodiment Layer that closes the perception, action, and reflection loop, and a Communicative Reasoning Layer that translates local reflections into goal-directed messages and post-dialog task updates. Detailed implementation is provided in Supplementary Sec. 1 (real world) and Sec. 2 (simulation).

LLM configuration. Unless otherwise specified, all LLM calls in our system use a unified configuration shared across robots and modules. In our experiments, we utilize the `gpt-4o-mini-2024-07-18` model. To balance creativity and determinism, we set the temperature to 0.7, while leaving all other hyperparameters at OpenAI API official default values. The core prompts are provided in Supplementary Sec. 3.

Embodiment layer. **Visual semantic perception**. The Visual Semantic Perception module enables each robot to interpret its surroundings by identifying visible objects, inferring their spatial relationships, and reasoning about which objects are misplaced and where they should be returned. At each time step, the robot acquires a visual observation and runs an object detector to identify objects. It then fuses the detections with depth information to estimate 3D object locations, projects them onto the map, and infers relative spatial relations among nearby objects (e.g., a clock on the table). This information supports efficient navigation and task-relevant spatial reasoning. Afterwards, for each newly detected object, we provide its relative spatial context to the LLM, which determines whether the object is misplaced and infers an appropriate target location. In our real-world deployment, we employ a hybrid perception scheme, using a fine-tuned YOLOv8 model to detect dynamic objects and fiducial markers (AR Tags) to precisely localize static receptacles. Spatial relationships are then inferred through geometric reasoning on the resulting 3D object locations. In contrast, our simulation environment utilizes the simulator's ground truth data, determining object visibility via ray casting and extracting spatial predicates from the scene graph directly. This allows us to isolate and evaluate the core decision-making algorithms, free from perception noise.

Action decision. The Action Decision module adopts a hierarchical design in which an LLM selects high-level robot skills conditioned on the robot's grounded state and the task allocation. At each decision cycle, we prompt the LLM to choose exactly one skill from a predefined library.

The skill library is instantiated separately for each platform to support heterogeneous teams. Each robot type exposes only the subset of the skill library that corresponds to its own capabilities. We define a basic set $\mathcal{B} = \{\texttt{Explore}, \texttt{Wait}, \texttt{Exit}\}$ that is shared across all platforms, providing common navigation and coordination capabilities. Building upon $\mathcal{B}$, the real-world robots used in this work implement the following platform specific skill sets: Go2 and Car use $\mathcal{B}$; AlienGo uses $\mathcal{B} \cup \{\texttt{GoPick}, \texttt{GoPlace}\}$; Stretch uses $\mathcal{B} \cup \{\texttt{GoPick}, \texttt{GoPlace}, \texttt{GoPull}\}$; and Cleaner uses $\mathcal{B} \cup \{\texttt{GoClean}\}$.

To encourage stable and conservative decision making, the prompt constrains the LLM to output a single skill name together with the required arguments (e.g., a target object identifier). Once a skill is selected, it is forwarded to the Action Execution module, which maps the high-level decision to an executable sequence of primitives and carries out the corresponding low-level control.

Action execution. The Action Execution module instantiates each high-level action decision as executable, low-level control commands that realize the selected skill. Given a skill output by the Action Decision module, it is first expanded into an ordered sequence of motion and manipulation action primitives using predefined, rule-based decompositions. The module then dispatches these primitives to the robot platform, monitors execution outcomes, and updates the robot's internal estimate of the environment's dynamic state accordingly. Controllers for individual primitives can be specified in advance using hand-designed controllers or learned from data via imitation learning (IL) and reinforcement learning (RL). The rule-

https://doi.org/10.1038/s41467-026-72797-8

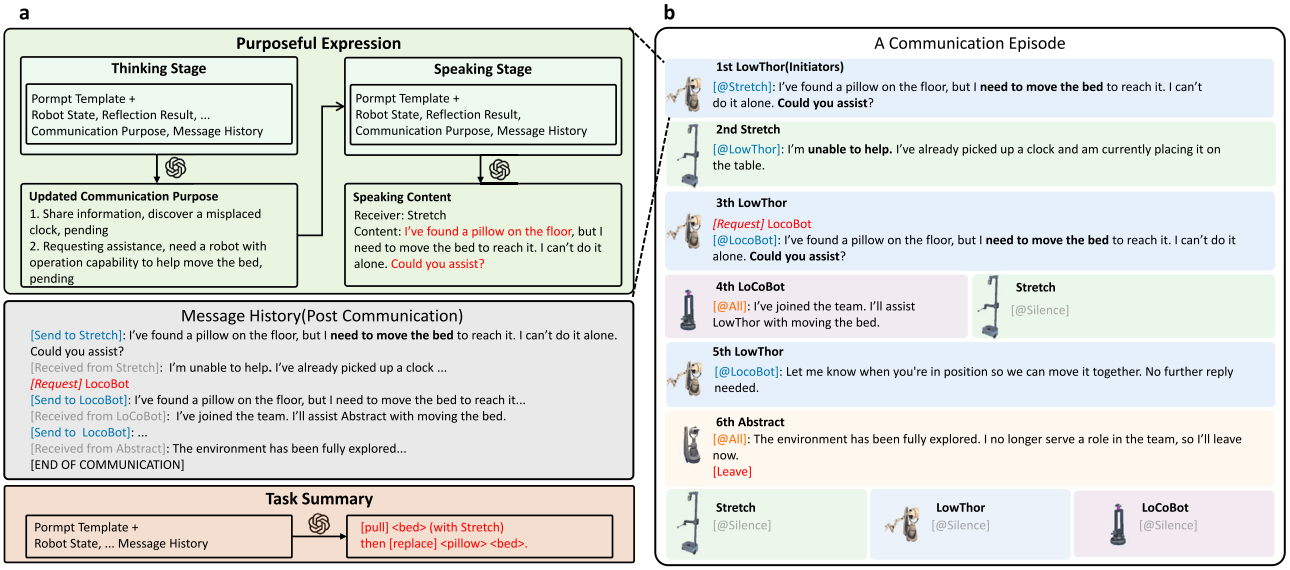


Fig. 10 | LowThor's Communicative Reasoning Layer in an autonomous need-driven dialog episode. a *LowThor*'s communicative reasoning pipeline during one episode managed by *Autonomous Need-Driven Communication*. **Top: Purposeful Expression**. The first utterance is generated in two stages: the *Thinking Stage* infers an explicit communication purpose from the prompt template, robot state, reflection signals, and dialog context. The *Speaking Stage* then produces the concrete, targeted message consistent with that purpose. **Middle: Message History**. After the episode ends, LowThor stores the full message history for consolidation. **Bottom: Task Summary**. LowThor summarizes the negotiated consensus and converts it into an executable local task plan. **b** An example episode of multi-round dialog. Multiple robots speak in sequence under *Autonomous Need-Driven Communication* and the interaction terminates naturally when no new responses are triggered. Each utterance is produced by the speaker's *Purposeful Expression* (Thinking → Speaking), and after termination, each robot independently runs *Task Summary* to update its own next task.

based skill-to-primitive mappings used in our implementation are as follows:

- `Explore⟨IDroom⟩` → `GoToObject⟨IDroom⟩` + `Explore⟨IDroom⟩`
- `GoPick⟨IDobject⟩` → `GoToObject⟨IDobject⟩` + `PickupObject⟨IDobject⟩`
- `GoPlace⟨IDobject⟩⟨IDreceptacle⟩` → `GoToObject⟨IDreceptacle⟩` + `PutObject⟨IDobject⟩⟨IDreceptacle⟩`
- `GoPull⟨IDobject⟩⟨direction⟩` → `GoToObject⟨IDobject⟩` + `PullObject⟨IDobject⟩⟨direction⟩`
- `GoClean⟨IDobject⟩` → `GoToObject⟨IDobject⟩` + `Clean⟨IDobject⟩`

For coordination, `Wait` invokes an `Idle` controller mode that holds position while continuing perception updates; `Exit` navigates to a predefined robot pool waypoint (`IDrobotpool`) and then enters `Idle`. In the real-world deployment, the execution stack is implemented using both ROS and ROS 2: AgileX LIMO and Unitree AlienGo run ROS Melodic, while Hello Robot Stretch, iRobot Create3, and Unitree Go2 run ROS 2 Foxy. Since ROS1 and ROS2 use different communication architectures, cross-robot coordination is implemented via a TCP-based centralized scheme over a fixed-IP LAN. Navigation is exposed through service interfaces, including mapping, localization, and planning provided by the robot manufacturer. Manipulation primitives such as picking and pulling are built on the open-source grasping pipelines and visual servoing with AR-tag-based localization. In simulation, the same primitives are mapped to the DynaTeamTHOR API. Navigation is performed with A* search over a NavMesh, and interactions such as picking are executed via kinematic attachment, enabling idealized and reproducible execution.

State assessment. The State Assessment module evaluates the success of an executed low-level action and the resulting state information, providing corresponding feedback when a failure occurs.

This module follows a two-stage pipeline that (i) deterministically determines whether an executed primitive action succeeds and produces a grounded, structured feedback signal, and (ii) diagnoses failures and suggests recovery via an LLM conditioned on this feedback.

After each low-level action, stage (i) performs a rule-based outcome check. It compares the pre-action and post-action states and evaluates simple geometric cues, such as relevant distances. It outputs a success flag. If the action fails, it additionally provides a coarse and verifiable failure reason.

In stage (ii), we provide the LLM with the structured outcome from stage (i) along with compact pre-action and post-action state summaries. Conditioned on these grounded signals, the LLM identifies the specific cause of failure and generates a concrete recovery strategy. The strategy is then either forwarded to the next action decision module or directly initiates communication. Based on this diagnosis, the robot decides whether teammate coordination is necessary. If coordination is required, it initiates a new round of team communication and integrates the reflection into its Purposeful Expression. Otherwise, it sends the recovery plan directly to the action decision module to regenerate a feasible plan.

Our state assessment is grounded in the platform's capabilities. In the real world, failure diagnosis is based on rich, hardware-level feedback. For instance, a grasp failure is diagnosed by analyzing force torque sensor readings for slippage, and navigation failures are assessed using fused LiDAR and IMU data. In simulation, assessment relies on logical checks against the simulator's ground-truth state. A grasp failure is determined by checking kinematic reachability, and navigation failures are identified by comparing the final position to the target coordinates. This dual approach ensures that our reasoning module receives physically meaningful feedback in the real world and logically consistent feedback in simulation.

Communicative reasoning layer. Purposeful expression. Purposeful Expression is the core module of the Communicative Reasoning Layer that generates targeted, structured messages. LLM-based natural

language communication in multi-robot teams often tends to result in off-topic, redundant, or irrelevant utterances, largely due to hallucinations and the lack of structured reasoning[68,69]. Such noise degrades coordination, causing misunderstandings, delays, and suboptimal execution. We attribute these failures to missing, explicit conversational goals that can be tracked across dialog rounds.

To address this issue, we design a two-stage pipeline: a Thinking Stage that infers communicative intent, and a Speaking Stage that generates the concrete utterance. This decoupling separates deciding what to say from deciding how to say it, improving interpretability and precision.

In the Thinking Stage, the LLM summarizes the robot state, reflection outputs, and message history into a set of communicative purposes (e.g., sharing information, allocating tasks, requesting assistance, offering help, or declining support). Purposes are maintained across rounds with explicit status labels (completed, pending, or abandoned), and new purposes can be introduced as task conditions evolve.

In the Speaking Stage, the robot verbalizes the pending purpose by selecting both the key content and the intended recipient(s), keeping the exchange focused and goal-directed. When the robot recognizes that the team's current capabilities are insufficient, it may issue Recruit$\langle Robot_n \rangle$ to recruit new members from the robot pool. The recruited robot joins the working team immediately.

Figure 10 illustrates how Purposeful Expression operates in a multi-round communication episode and how the LLM is used throughout this process. In Fig. 10a, we visualize LowThor's Communicative Reasoning Layer during an episode managed by Autonomous Need-Driven Communication. Specifically, the top panel (Purposeful Expression) shows how LowThor generates its first utterance using the LLM in two stages: the Thinking Stage infers an explicit communicative Purpose from the prompt template, robot state, reflection signals, and dialog context. Conditioned on this Purpose, the Speaking Stage then uses the LLM to produce the concrete, recipient-targeted Speaking Content. Fig. 10b shows a full communication episode: multiple robots are scheduled to speak in sequence under Autonomous Need-Driven Communication, and the episode terminates naturally when no further responses are triggered. This highlights the relationship between the two mechanisms: Autonomous Need-Driven Communication organizes who speaks when and when the dialog ends, while each individual utterance is generated by the current speaker's Purposeful Expression (Thinking → Speaking), as exemplified by LowThor in Fig. 10a.

Task summary. The Task Summary module consolidates the outcome of a communication episode into an explicit, executable update of each robot's local plan. After a need-driven multi-round communication terminates, the robot reviews the full message history and extracts the team-level consensus, including (i) updated task assignments, (ii) team composition changes (e.g., Recruit/Release), and (iii) key coordination commitments (e.g., who will push, wait, or navigate to a location). It then rewrites its own current task. By grounding negotiated consensus into a concise, agent-local task. Task Summary ensures that all participants transition to consistent next actions without requiring a centralized controller.

Autonomous need-driven communication

While Purposeful Expression determines *what* a robot should say and generates a single targeted utterance, Autonomous Need-Driven Communication orchestrates the entire communication process. When a robot detects a coordination need and initiates communication, this mechanism manages the subsequent multi-round dialog until an actionable consensus is reached. In doing so, it provides the coordination backbone that transforms individual, purpose-driven utterances into team-level agreements: it bridges individual robots and global coordination by structuring multi-round interactions in which task allocation, coordination plans, and recruitment decisions are jointly produced through thorough discussion.

We model the multi-round dialog as a breadth-first traversal (BFS) over a dynamically evolving communication graph $G = (V, E)$, where vertices V are the participating robots and edges E represent directed messages. The process, detailed in Algorithm 1, begins with a set of seed speakers(the robots that initiate communication) $V_{\text{seed}} \subseteq V$. In each round, a current speaker $v \in V$ generates an utterance and explicitly names a set of target robots V_{target}. These robots are added to the graph, and an edge is created from v to each $v' \in V_{\text{target}}$. The protocol schedules these new robots to speak in subsequent rounds, thereby expanding the graph on demand. The dialog terminates naturally when no new robots are addressed, signifying that a consensus or saturation point has been reached.

To ensure efficiency and robustness, the protocol incorporates several lightweight pruning and safety mechanisms, including a turn limit on pairwise conversations to avoid deadlocks. This graph-based approach provides a flexible yet structured mechanism for decentralized team formation and task allocation, allowing complex coordination plans to be negotiated without a central controller.

Algorithm 1. Autonomous Need-Driven Communication

Require: Overall robot pool $\mathcal{R} = \{R^{(1)}, R^{(2)}, \ldots, R^{(N)}\}$, robot team $\mathcal{T} = \{T^{(1)}, T^{(2)}, \ldots, T^{(M)}\} \subseteq \mathcal{R}$, robots intend to speak $\mathcal{K} = \{K^{(1)}, K^{(2)}, \ldots, K^{(L)}\} \subseteq \mathcal{T}$

Constants:

PM: Maximum communications for each robot pair

TM: Maximum total communications

1: Initialization

count$(R^{(i)}, R^{(j)}) \leftarrow 0, \forall (R^{(i)}, R^{(j)}) \in \mathcal{R} \times \mathcal{R}$, $R^{(i)} \neq R^{(j)}$

total_count ← 0

$\mathcal{K} \leftarrow$, queue$(\mathcal{K})$ ▷Convert to FIFO queue

2: **while** $\mathcal{K} \neq \varnothing$ **and** total_count < TM

3: $K^{(i)} \leftarrow$ dequeue $(\mathcal{K})$

4: receivers, contents ← message_generation($K^{(i)}$) contents (corresponding messages)

$R^{(i)} \in \mathcal{R}$ corresponding to $K^{(i)} \in \mathcal{K}$

5: $R^{(i)}$.add_memory(t, receivers, contents)

6: **for** each $(R^{(j)}, c) \in$ (receivers, contents)

7: $R^{(j)}$.add_memory(t, R^(i), c)

8: **if** (count$(R^{(i)}, R^{(j)}) <$ PM) and $R^{(j)} \notin \mathcal{K}$

9: enqueue$(\mathcal{K}, R^{(j)})$

10: count$(R^{(i)}, R^{(j)}) \leftarrow$ count$(R^{(i)}, R^{(j)}) + 1$

11: count$(R^{(j)}, R^{(i)}) \leftarrow$ count$(R^{(i)}, R^{(j)})$

12: total_count ← total_count + 1

13: **end if**

14: **end for**

15: **end while**

Evaluation metrics

To comprehensively evaluate the performance of different collaboration strategies and algorithmic components, we define a unified set of metrics applicable to both real-world and simulated environments. These metrics capture task completion outcomes, coordination efficiency, team adaptability, and resource utilization.

Evaluation protocol. We denote N_{epi} as the total number of experimental episodes. Unless otherwise specified, we report the *mean* value of each metric over episodes: $\overline{X} = \frac{1}{N_{\text{epi}}} \sum_{i=1}^{N_{\text{epi}}} X_i$, where X_i is the value computed in episode i. For time-indexed quantities in an episode, we use $t \in \{1, \ldots, \text{TS}_i\}$ to denote the temporal step index, and $\text{TM}_{i,t}$ to denote the number of active robots (team members) at step t.

Metrics. In real-world experiments, we report a compact subset due to physical constraints, while simulation additionally reports fine-grained metrics. We define all major metrics below for completeness.

See the Supplementary Sec. 4 for auxiliary metrics and extended analysis.

- Success Rate (SR). The strict task success indicator. $SR_i = 1$ if all target objects are placed in reasonable locations in episode i, otherwise $SR_i = 0$. We report $\%SR = 100 \cdot \overline{SR}$.
- Partial Success Rate (PS, simulation only). $PS_i = \frac{K_i^{suc}}{K_i}$, where K_i is the number of objects that need to be rearranged in episode i, and K_i^{suc} is the number successfully placed in reasonable locations. We report $\%PS = 100 \cdot \overline{PS}$.
- Task Completion Time / Temporal Steps (TS). TS_i is the number of temporal steps until termination in episode i (successful termination or timeout). We report $\#TS = \overline{TS}$ (smaller is better).
- Total Action Steps (AS, simulation only). AS_i is the sum of action steps executed by all robots in episode i (including steps until timeout if unsuccessful). We report $\#AS = \overline{AS}$ (smaller implies lower energy/effort).

Data availability

The datasets generated in this study, along with the source data underlying the figures and tables, have been deposited in the Zenodo database under DOI 10.5281/zenodo.19240700 (https://doi.org/10.5281/zenodo.19240700)[70].

Code availability

The code used to reproduce the results of this study, along with the source code underlying the figures and tables, has been deposited in the Zenodo database under DOI 10.5281/zenodo.19240700 (https://doi.org/10.5281/zenodo.19240700)[70].

References

1. Rizk, Y., Awad, M. & Tunstel, E. W. Cooperative heterogeneous multi-robot systems: a survey. *ACM Comput. Surv. (CSUR)* **52**, 1–31 (2019).
2. Darmanin, R. N. & Bugeja, M. K. *A review on multi-robot systems categorised by application domain*. In *25th Mediterranean Conference on Control and Automation, MED 2017, Valletta, Malta, July 3-6, 2017*,701–706 (IEEE, 2017).
3. Howard, A., Parker, L. E. & Sukhatme, G. S. Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection. *Int. J. Robot. Res.* **25**, 431–447 (2006).
4. Ju, C. & Son, H. I. Modeling and control of heterogeneous agricultural field robots based on ramadge-wonham theory. *IEEE Robot. Autom. Lett.* **5**, 48–55 (2020).
5. Schuster, M. J. et al. The ARCHES space-analogue demonstration mission: Towards heterogeneous teams of autonomous robots for collaborative scientific sampling in planetary exploration. *IEEE Robot. Autom. Lett.* **5**, 5315–5322 (2020).
6. Sharma, V. et al. CH-MARL: a multimodal benchmark for cooperative, heterogeneous multi-agent reinforcement learning. Preprint at https://arxiv.org/abs/2208.13626 (2022).
7. Sudarshan, S. K. V. et al. A heterogeneous robotics team for large-scale seismic sensing. *IEEE Robot. Autom. Lett.* **2**, 1328–1335 (2017).
8. Stone, P. & Veloso, M. Multiagent systems: a survey from a machine learning perspective. *Autonomous Robots* **8**, 345–383 (2000).
9. Gerkey, B. P. & Matarić, M. J. A formal analysis and taxonomy of task allocation in multi-robot systems. *Int. J. Robot. Res.* **23**, 939–954 (2004).
10. Lissandrini, N., Verginis, C. K., Roque, P., Cenedese, A. & Dimarogonas, D. V. Decentralized nonlinear MPC for robust cooperative manipulation by heterogeneous aerial-ground robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2020, Las Vegas, NV, USA*, 1531–1536 (IEEE, 2020).
11. Yue, Y., Wen, M., Putra, Y., Wang, M. & Wang, D. Tightly-coupled perception and navigation of heterogeneous land-air robots in complex scenarios. In *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China*,10052–10058 (IEEE, 2021).
12. Haldane, D. W., Fankhauser, P., Siegwart, R. & Fearing, R. S. Detection of slippery terrain with a heterogeneous team of legged robots. In *2014 IEEE International Conference on Robotics and Automation, ICRA 2014, Hong Kong, China*, 4576–4581 (IEEE, 2014).
13. Wallace, D., He, Y. H., Vaz, J. C., Georgescu, L. & Oh, P. Y. Multimodal teleoperation of heterogeneous robots within a construction environment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2020, Las Vegas, NV, USA*, 2698–2705 (IEEE, 2020).
14. Dai, W., Bidwai, A. & Sartoretti, G. Dynamic coalition formation and routing for multirobot task allocation via reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 16567–16573 (2024).
15. Stone, P., Kaminka, G. A., Kraus, S. & Rosenschein, J. S. Ad hoc autonomous agent teams: Collaboration without pre-coordination. *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*, 1504–1509 (AAAI Press, 2010).
16. Barrett, S., Rosenfeld, A., Kraus, S. & Stone, P. Making friends on the fly: Cooperating with new teammates. *Artif. Intell.* **242**, 132–171 (2017).
17. Mirsky, R. et al. A survey of ad hoc teamwork: Definitions, methods, and open problems. Preprint at https://arxiv.org/abs/2202.10450 (2022).
18. Wang, J., Li, Y., Zhang, Y., Pan, W. & Kaski, S. Open ad hoc teamwork with cooperative game theory. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, vol.235 (PMLR, 2024).
19. Liu, X., Li, P., Yang, W., Guo, D. & Liu, H. Leveraging large language model for heterogeneous ad hoc teamwork collaboration. *Robotics: Science and Systems XX, Delft, The Netherlands*,(2024).
20. Jones, E. G. et al. Dynamically formed heterogeneous robot teams performing tightly-coordinated tasks. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, ICRA*,570–575 (IEEE, 2006).
21. Gunn, T. & Anderson, J. Dynamic heterogeneous team formation for robotic urban search and rescue. *J. Comput. Syst. Sci.* **81**, 553–567 (2015).
22. Rahman, M., Höpner, N., Christianos, F. & Albrecht, S. V. Towards open ad hoc teamwork using graph-based policy learning. In *International Conference on Machine Learning*, 8776–8786 (PMLR, 2021).
23. Schmutz, J. B. & Eppich, W. J. Helping fluid teams work: a research agenda for effective team adaptation in healthcare. *Acad. Med.* **94**, 1269–1273 (2019).
24. Driskell, T., Funke, G., Tolston, M., Capiola, A. & Driskell, J. E. Fluid and virtual teams. *Small Group Res.* **55**, 680–705 (2024).
25. Driskell, T., Funke, G., Tolston, M. T., Capiola, A. & Driskell, J. Supporting fluid teams: a research agenda. *Front. Psychol.* **15**, 1327885 (2024).
26. Winters, B. D. et al. Rapid response systems: a systematic review. *Crit. Care Med.* **35**, 1238–1243 (2007).
27. Entin, E. E. & Serfaty, D. Adaptive team coordination. *Hum. Factors* **41**, 312–325 (1999).
28. Hettinger, A. Z. et al. Ten best practices for improving emergency medicine provider–nurse communication. *J. Emerg. Med.* **58**, 581–593 (2020).
29. Woolley, A. W., Chabris, C. F., Pentland, A., Hashmi, N. & Malone, T. W. Evidence for a collective intelligence factor in the performance of human groups. *Science* **330**, 686–688 (2010).
30. Woolley, A. W., Aggarwal, I. & Malone, T. W. Collective intelligence and group performance. *Curr. Dir. Psychol. Sci.* **24**, 420–424 (2015).

31. Janssens, M., Meslec, N. & Leenders, R. T. A. Collective intelligence in teams: contextualizing collective intelligent behavior over time. *Front. Psychol.* **13**, 989572 (2022).
32. Tan, S., Ge, M., Guo, D., Liu, H. & Sun, F. Knowledge-based embodied question answering. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**, 11948–11960 (2023).
33. Tellex, S., Knepper, R. A., Li, A., Rus, D. & Roy, N. Asking for help using inverse semantics. *Robotics: Science and Systems X, University of California, Berkeley, USA, 12–16,*(2014).
34. Srinivasan, V. & Takayama, L. Help me please: Robot politeness strategies for soliciting help from humans. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, May 7-12, 2016*, 4945–4955 (ACM, 2016).
35. Singh, K. P. et al. Ask4help: Learning to leverage an expert for embodied tasks. *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA,* (2022).
36. Ren, A. Z. et al. Robots that ask for help: Uncertainty alignment for large language model planners. *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, vol. 229 of *Proceedings of Machine Learning Research*, 661–682 (PMLR, 2023).
37. Liang, K., Zhang, Z. & Fisac, J. F. Introspective planning: Aligning robots' uncertainty with inherent task ambiguity. *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10–15, 2024* (2024).
38. Shah, R., Yu, A., Zhu, Y., Zhu, Y. & Martín-Martín, R. Bumble: Unifying reasoning and acting with vision-language models for building-wide mobile manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*,13337–13345 (2025).
39. Li, H., Zhang, S. & Guo, D. Robocleaner: Robotic tabletop cleaning via vlm-powered multi-agent collaboration. *IEEE Trans. Autom. Sci. Eng.* **23**, 3651–3660 (2026).
40. Chen, J. et al. EMOS: embodiment-aware heterogeneous multi-robot operating system with LLM agents. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025* (OpenReview.net, 2025).
41. Liu, K. et al. COHERENT: collaboration of heterogeneous multi-robot system with large language models. Preprint at https://arxiv.org/abs/2409.15146 (2024).
42. Yu, W. et al. MHRC: closed-loop decentralized multi-heterogeneous robot collaboration with large language models. Preprint at https://arxiv.org/abs/2409.16030 (2024).
43. Zhang, H. et al. Building cooperative embodied agents modularly with large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024* (OpenReview.net, 2024).
44. Sun, N. et al. AssistantX: An llm-powered proactive assistant in collaborative human-populated environment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2025, Hangzhou, China, October 19-25, 2025*, 3352–3359 (IEEE, 2025).
45. Mon-Williams, R., Li, G., Long, R., Du, W. & Lucas, C. G. Embodied large language models enable robots to complete complex tasks in unpredictable environments. *Nat. Mach. Intell.* **7**, 592–601 (2025).
46. Liang, J. et al. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation, ICRA 2023, London, UK, May 29 - June 2, 2023*, 9493–9500 (IEEE, 2023).
47. Huang, W. et al. Voxposer: Composable 3d value maps for robotic manipulation with language models. *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, **229** *Proceedings of Machine Learning Research*, 540–562 (PMLR, 2023).
48. Meng, Y. et al. Preserving and combining knowledge in robotic lifelong reinforcement learning. *Nat. Mach. Intell.* **7**, 256–269 (2025).
49. Hong, S. et al. MetaGPT: Meta programming for A multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024* (OpenReview.net, 2024).
50. Qian, C. et al. ChatDev: Communicative agents for software development. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, 15174–15186 (Association for Computational Linguistics, 2024).
51. Park, J. S. et al. Generative agents: Interactive simulacra of human behavior. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, UIST 2023, San Francisco, CA, USA, 29 October 2023- 1 November 2023*, 2:1–2:22 (ACM, 2023).
52. Liu, H., Guo, D. & Huang, K. Learning for embodiment and embodiment for learning. *Nat. Rev. Electr. Eng.* **2**, 651–653 (2025).
53. Liu, H., Guo, D. & Cangelosi, A. Embodied intelligence: A synergy of morphology, action, perception and learning. *ACM Comput. Surv.* **57**, 1–36 (2025).
54. Sarch, G. et al. TIDEE: tidying up novel rooms using visuo-semantic commonsense priors. *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXIX*, vol. 13699 of *Lecture Notes in Computer Science*, 480–496 (Springer, 2022).
55. Liu, X., Guo, D., Zhang, X. & Liu, H. Heterogeneous embodied multi-agent collaboration. *IEEE Robot. Autom. Lett.* **9**, 5377–5384 (2024).
56. Kolve, E. et al. AI2-THOR: An interactive 3d environment for visual ai. Preprint at https://arxiv.org/abs/1712.05474 (2017).
57. Wang, C., Du, B., Xu, J. & Liu, H. Demonstrating HumanTHOR: A simulation platform and benchmark for human-robot collaboration in a shared workspace. *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024* (2024).
58. Wu, Q. et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation framework. *First Conference on Language Modeling* (2024).
59. Guo, X. et al. Embodied LLM agents learn to cooperate in organized teams. *IEEE Trans. Comput. Soc. Syst.* **13**, 2514–2530 (2026).
60. Shapiro, D. L. & Brett, J. M. Comparing three processes underlying judgments of procedural justice: A field study of mediation and arbitration. *J. Pers. Soc. Psychol.* **65**, 1167 (1993).
61. Conlon, D. E., Moon, H. & Ng, K. Y. Putting the cart before the horse: The benefits of arbitrating before mediating. *J. Appl. Psychol.* **87**, 978 (2002).
62. Jackson, J. W. Realistic group conflict theory: A review and evaluation of the theoretical and empirical literature. *Psychol. Rec.* **43**, 395 (1993).
63. Sherif, M. et al. Intergroup Conflict and Cooperation: The Robbers Cave Experiment. (University Book Exchange Norman, OK, 1961).
64. Omoto, A. M. & Snyder, M. Sustained helping without obligation: motivation, longevity of service, and perceived attitude change among AIDS volunteers. *J. Pers. Soc. Psychol.* **68**, 671 (1995).
65. Mowen, J. C. & Sujan, H. Volunteer behavior: A hierarchical model approach for investigating its trait and functional motive antecedents. *J. Consum. Psychol.* **15**, 170–182 (2005).
66. Cialdini, R. B. & Goldstein, N. J. Social influence: Compliance and conformity. *Annu. Rev. Psychol.* **55**, 591–621 (2004).
67. Cialdini, R. B. & Trost, M. R. Social influence: social norms, conformity and compliance. *The Handbook of Social Psychology* 4th edn, pp. 151–192 (McGraw-Hill, 1998).
68. Gunjal, A., Yin, J. & Bas, E. Detecting and preventing hallucinations in large vision language models. *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, 18135–18143 (AAAI Press, 2024).

69. He, Y. et al. Evaluating and mitigating object hallucination in large vision-language models: Can they still see removed objects? *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA,* 6841–6858 (Association for Computational Linguistics, 2025).
70. Li, P., Zhang, S., Yang, W., Hu, Y., Wang, Y., Wang, D., Li, H., Guo, D. & Liu, H. Data and code for: Proactive Collaboration via Autonomous Interaction. *Zenodo* https://doi.org/10.5281/zenodo.19240700 (2026).

Acknowledgements

This work was funded by the National Key Research and Development Program of China [grant number 2024YFB4708000], the National Natural Science Fund [grant numbers 62273054, 62025304, and 62120106005], and the Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China [grant number JYB2025XDXM109].

Author contributions

H.L. and D.G. lead the project. P.L. and S.Z. designed and implemented the main algorithms and performed the data analysis. P.L., S.Z., Y.H. and D.W. developed the simulation environment and conducted the corresponding experiments. W.Y. led the physical experiments, and along with P.L., S.Z., Y.H., Y.W., D.W. and H.Li, carried out the physical trials. P.L., S.Z., W.Y., D.G. and H.L. co-wrote the paper and prepared the illustrations. All authors discussed the results and commented on the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-026-72797-8.

Correspondence and requests for materials should be addressed to Di Guo or Huaping Liu.

Peer review information *Nature Communications* thanks the anonymous reviewers for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2026