Communicative PARTNR: Natural Language Communication under Partial Observability in Human–Robot Collaboration

HoBeom Jeon[1] Hyungmin Kim[1] Dohyung Kim[1,2†] Minsu Jang[1,2] Jaehong Kim[2]
[1]University of Science and Technology, Deajeon, Republic of Korea
[2]Electronics and Telecommunications Research Institute, Deajeon, Republic of Korea
{tiger, khm159, dhkim008, minsu, jhkim504}@etri.re.kr

Abstract

Natural language communication is a key mechanism for coordination in human–robot collaboration under partial observability, where agents possess only local views of the environment. While prior work often assumes fully reliable or unconstrained message channels, real-world settings impose delivery uncertainty and range limitations that complicate when and how agents should communicate. We present Communicative PARTNR, an extension of the PARTNR benchmark that introduces a range-limited channel and four levels of system feedback information: Opaque (no confirmation), Binary (success/failure), Causal (failure reason), and Traceable (failure reason with partner state). Using LLM-based embodied agents in decentralized household tasks, we find that minimal and unclear feedback (Opaque) yields higher task success and completion rates than richer alternatives, despite generating fewer and shorter dialogues. Analysis reveal that excessive detail can divert agent reasoning from task execution, whereas concise system feedback maintain focus and coordination efficiency. These results underscore the importance of designing dialogue strategies and context representations that enable agents to exploit communication outcome information effectively without incurring unnecessary cognitive or temporal overhead. Code is available in https://github.com/HoBeom/Communicative-PARTNR.

1. Introduction

In human-robot collaborative scenarios, each agent typically perceives only a slice of the world. For example, a domestic robot may have camera views of one room while the human partner is in another, so neither has the full picture. Such partial observability fundamentally complicates coordination, as agents must act with incomplete information. Humans address this in teamwork by communicating: they ask questions, share relevant observations, and explain their intentions. Enabling robots to engage in similar free-form natural language communication with humans is a tantalizing goal that promises more flexible and general collaboration. However, it also raises challenges in interpretation, generation, and deciding what to communicate and when, especially under uncertainty.

Recent benchmarks like PARTNR (Planning and Reasoning Tasks in humaN-Robot collaboration) [2] have begun to target these issues. PARTNR is a large-scale suite of simulated household tasks for a human and a robot agent, specified in natural language. Tasks involve spatial and temporal constraints and require dividing responsibilities between agents. Crucially, scenarios can be configured with partial observability to test decentralized cooperation. Initial studies on PARTNR reveal that even powerful LLM-based planners struggle at collaborative consistency: they often exhibit poor coordination, failing to track the human partner's actions or to recover from errors. Human-in-the-Loop evaluations further show that, despite achieving high success rates in some settings, LLM agents contribute far less to the shared workload than their human partners (e.g., only 16% task offloading for Human-ReAct vs. 50% ideal balance) [2]. This imbalance forces humans to shoulder the majority of the task execution, slowing overall progress and highlighting the need for richer, well-timed communication strategies that enable robots to take on a fairer share of collaborative work.

Communication is an obvious avenue to bridge this gap. By exchanging information, a robot can overcome its limited viewpoint, and a human can better understand the robot's intentions. Yet, naive implementations of free-form agent chat can be problematic. Unstructured dialogues between agents may become inefficient or even counterproductive, as irrelevant or misleading utterances (so-called "meaningless chatter") can lead to confusion or cascading hallucinations [12]. Indeed, recent multi-agent studies have noted that unrestricted LLM agents conversing without constraints or protocols can veer off track. The challenge, then,

† Corresponding author

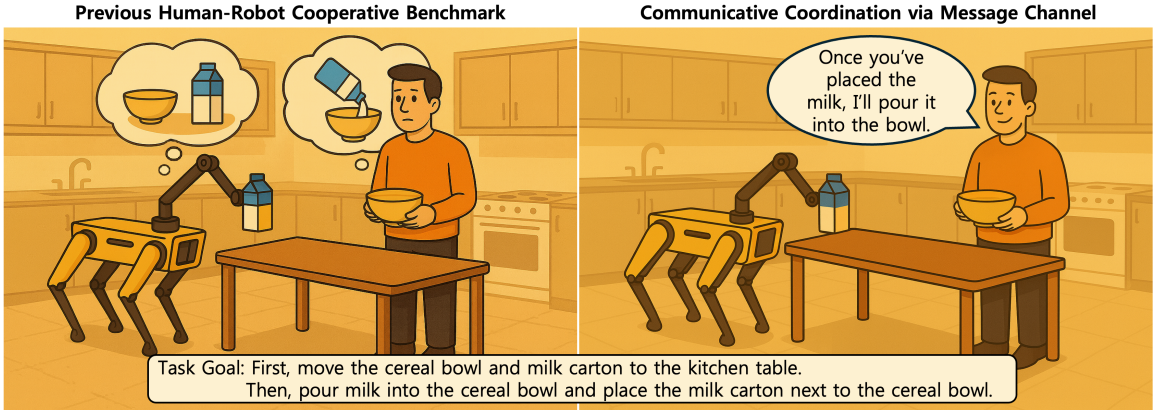


Figure 1. Non-communicative vs. Communicative Human–Robot Collaboration under Partial Observability. Communicative coordination mitigates conditional deadlock in partially observable human–robot collaboration. In the non-communicative setting (left), both agents wait indefinitely due to mutually dependent goals, leading to deadlock. In the communicative setting (right), a brief utterance establishes common ground and task order, enabling completion of the shared objective. (Illustration generated using AI tools)

is to harness the expressiveness of natural language while maintaining grounding and relevance in communication.

In this work, we tackle the problem of natural language communication for human-robot collaboration under partial observability. We introduce Communicative PARTNR, an extension of the PARTNR framework where the robot is endowed with conversational abilities, and we systematically study how different system feedback policies influence collaboration. We consider four modes that vary in delivery transparency: Opaque (no confirmation of delivery), Binary (success/failure only), Causal (failure reason provided), and Traceable (failure reason plus partner state). This design allows us to isolate the impact of feedback granularity on coordination performance, highlighting that more informative feedback does not necessarily lead to better outcomes under spatial and perceptual constraints.

2. Related Work

Multi-agent Coordination under Partial Observability. Coordinating agents with incomplete information is a central challenge in cooperative AI. Decentralized partially observable Markov decision processes (Dec-POMDPs) formalize this setting and highlight its intractability [1]. Recent work has shifted toward realistic multi-agent benchmarks that explicitly evaluate coordination under uncertainty. The PARTNR benchmark [2] is one such example, featuring human–robot household tasks where agents operate in separate rooms with only local observations. These tasks can be configured in centralized settings, where a single high-level planner controls both agents and has access to their combined state, or decentralized settings, where each agent independently plans from its own partial view and must reason about the partner's behavior. This distinction captures the coordination overhead introduced by limited observability. Building on this line of work, EMOS [3] addresses heterogeneous robot collaboration by introducing a centralized discussion phase in which all robots exchange high-level action intents to build a joint plan. Once the plan is established, each robot switches to a decentralized execution phase, performing only its assigned subtasks. This structure improves interoperability across heterogeneous agents and reduces the need for continuous message exchange during execution, while still benefiting from global plan alignment achieved in the initial discussion.

LLM Agents in Dialogue. The emergence of large language models has led to growing interest in using them as decision-making and communication modules for multiple agents. CAMEL [6] demonstrated that two or more LLM agents can be assigned complementary roles and achieve shared goals by engaging in goal-directed natural language dialogue, enabling dynamic plan alignment, adaptation to changes, and transparent reasoning. MetaGPT [4] extended this paradigm to software engineering by coordinating specialized LLM agents (e.g., product manager, engineer, tester) that collaborate through structured conversation to deliver complete projects. Beyond small teams, Generative Agents [9] modeled an interactive town of 25 LLM-powered characters whose conversations and collaborations produced emergent social behaviors resembling real-world dynamics. Collectively, these works illustrate the potential of natural language as a unifying medium for cooperation, enabling multi-agent systems to develop emergent strategies and maintain interpretable reasoning processes across a range of collaborative scenarios.

Natural Language in Embodied Collaboration. On the human–robot collaboration front, CoELA (Cooperative Embodied Language Agent) [13] provides a notable example of an LLM-driven partner. It places a human and a

robot in a simulated home with an always-available chat channel, enabling tasks like Communicative Watch-And-Help and TDW-MAT. In this fully connected setting, the robot agent frequently communicates its status and intentions; however, unfettered dialog can become inefficient. Users observed that the CoELA agent sometimes produced verbose status reports or planning monologues that did not directly help the human partner. To address such issues, recent works propose more structured communication. Cooperative Plan Optimization (CaPo) [8] inserts a dedicated multi-turn planning phase before execution: all agents collaboratively formulate a high-level meta-plan through discussion, with one agent proposing the plan and others providing feedback, until consensus is reached. Another approach, REVECA [11], leverages a relevance-filtered LLM architecture with adaptive planning and trajectory validation to keep conversations focused and avoid false assumptions. Both CaPo and REVECA demonstrate that constraining and guiding dialogue can boost efficiency, but they still assume an ideal communication channel where agents can message each other freely at any time. In fact, these methods allow an almost oracle-like exchange of information (e.g. continuously sharing each agent's location, action history, and state), which can lead to significant communication overhead [7]. In contrast to prior work, we investigate human–robot collaboration in a more realistic setting characterized by partial communication and asymmetric action capabilities. Unlike fully connected benchmarks, our environment constrains message exchange to situations where the partners are physically close enough for the signal to be received, and the household robot is unable to perform certain operations that require human intervention. These constraints make both the timing and the content of communication critical for maintaining coordination.

3. Communicative PARTNR Benchmark

3.1. Problem Setting

We consider a collaborative household task environment where a human and a robot work together to achieve a common goal. The tasks are drawn from the PARTNR benchmark, which consists of realistic home activities (e.g., "Help me move the plant from the bedroom to the living room.") designed to evaluate human-robot coordination in simulated multi-room houses. The environment is simulated in Habitat [10] and populated with the Habitat Synthetic Scenes Dataset (HSSD) [5] of household objects. Each agent perceives only a limited egocentric view and cannot directly access the partner's observations. The robot lacks specific abilities such as cleaning, turning appliances on or off, and pouring liquids, and must request the human's assistance when these actions are required. Communication is proximity-limited so that spoken messages are only delivered if the listener is within range. This combination of perceptual and communicative constraints necessitates careful coordination strategies to ensure all task goals are completed efficiently.

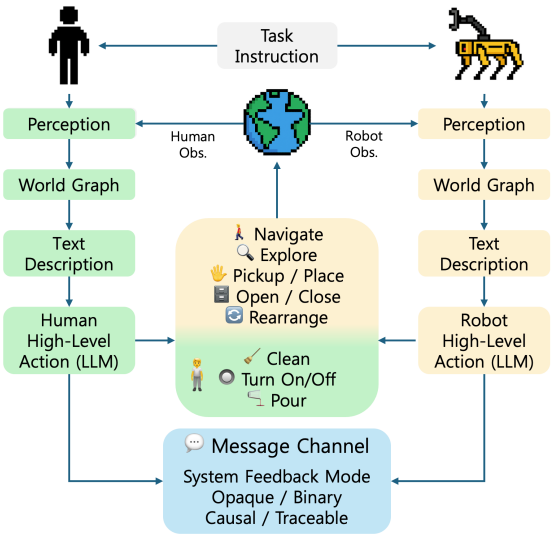


Figure 2. Overview of the Communicative PARTNR architecture. Two embodied LLM agents operate under partial observability, each maintaining an independent world graph and high-level action planning. The human agent possesses three additional capabilities: Clean, Turn On/Off, and Pour, in addition to navigation and manipulation primitives. A partial message channel enables natural language exchange within range, with four system feedback modes (Opaque, Binary, Causal, Traceable) shaping coordination strategies.

We model the collaborative task as a decentralized planning problem with optional communication. At each planning step, the human and robot each receive their own observation (e.g., detected object list, delivered messages) and then choose a high-level action. Human actions include high-level instructions (e.g., clean, turn on, pour), while the robot's actions consist of navigation and manipulation primitives (e.g., move, pick, place, open/close). In addition, both agents can invoke a communicate action to send a natural language message to the other. There is no central controller; instead, the human and robot plan their actions based on their individual observations and the shared dialogue history. The objective is to fulfill all goal conditions of the task collaboratively. We evaluate our approach on tasks from PARTNR's validation set, which comprises 12 house scenes containing varied objects and a total of 1,000 task instructions. This setup enables us to analyze whether adding a communication channel helps resolve coordination failures (such as action timing conflicts or "race conditions") observed in prior non-communicative configurations.

3.2. LLM-based Embodied Agent

The embodied agent is controlled by a Large Language Model (LLM) that maps dialogue and the perceptual context represented by the World Graph into actions. At each decision point, the LLM receives a structured textual description containing the agent's task instructions, available high-level actions, memory context of past actions, and current observations. Based on this input, the LLM generates an output that is parsed into either a physical action command or a communication act. Our design follows a ReAct-style approach, in which the LLM prompt maintains a chain-of-thought and action history from previous steps. In practice, the prompt-based agent simply incorporates all prior "think–act–observation" traces as contextual information for the next reasoning cycle. Any message from the partner agent is appended to the observation description. For example, if the human agent invokes the command `SendMessageTool[Please bring the backpack to the bedroom and place it on the bed]`, the robot agent's observation text will include a corresponding line such as `Message from Agent_2: "Please bring the backpack to the bedroom and place it on the bed"`.

We employ the Llama-3.3 model with 70B parameters as the agent controller, selected for its strong reasoning capabilities. Preliminary experiments indicated that it outperformed smaller variants as well as the previously used Llama-3.1 model on PARTNR tasks, both of which were evaluated for comparison. A central component of the agent is the system prompt, a fixed prefix to the LLM input that can enforce a specific communication style or strategy. By modifying the system prompt, the robot's communication mode can be altered (e.g., verbose versus succinct, inquiry-first versus act-first). In this study, however, the system prompt is kept consistent and neutral across all experiments to isolate the effects of external system feedback rather than relying on handcrafted dialogue policies.

To enable explicit communication between agents, we extend the Habitat action API with a `SendMessage` tool. This tool allows the LLM planner to produce a natural language utterance as an explicit action, with the content enclosed in square brackets (e.g., `SendMessageTool[Message content]`). By including this tool in the LLM's action space, the agent can decide at any timestep whether to execute a physical operation or send a message, thereby facilitating natural coordination with the human partner. All other available actions follow those in the original PARTNR benchmark (e.g., `Navigate`, `Explore`, `Open`, `Rearrange`). The simulated human agent is also implemented as an LLM-based agent, with a distinct prompt reflecting human capabilities and perspective, following prior work [13]. This human proxy executes high-level instructions and engages in dialogue, enabling controlled and repeatable experiments. As future work, we plan to replace this simulated human with real participants for further validation.

3.3. System Feedback Strategies

In realistic scenarios, message delivery is not guaranteed; for instance, the human partner may be too far away or located in a different room. To examine the role of communication transparency, we introduce system feedback modes that vary the information provided to the robot regarding the delivery status of its sent messages. Specifically, after the robot uses the `SendMessage` action, the environment returns a feedback message indicating delivery success or failure, with the level of detail determined by the feedback mode. We consider four levels:

- **Opaque**: The robot receives no confirmation of delivery. The system returns a vague acknowledgement (e.g., "Message sent. Delivery status unknown."), forcing the robot to proceed under uncertainty. This represents a highly lossy channel where confirmation is entirely absent.
- **Binary**: The robot is informed only whether the message was successfully delivered (e.g., "Message delivered." / "Message failed."). No explanation for failure is given, requiring the robot to infer potential causes and adapt without explicit guidance.
- **Causal**: In addition to success/failure, the robot receives a brief reason for failure. For example, "Message failed (partner not in the same room or beyond 7.0m range)." Such contextual information allows the robot to take corrective action, such as moving closer before retrying communication.
- **Traceable**: The robot receives the delivery status, the failure reason, and additional partner state information. For example, "Message failed (not in range). Partner is in 'hallway1', approximately 8.3m away." This richest mode enables highly targeted adjustments, such as navigating directly to the partner's location to re-establish communication.

The only difference between these modes is the informativeness of the feedback message; the LLM's base planning prompt and policy remain unchanged. The robot is not explicitly instructed on how to respond to communication uncertainty, allowing us to observe emergent adaptation strategies under varying levels of feedback transparency.

4. Experiments and Results

4.1. Evaluation Metrics.

We assess performance across four complementary metrics:

(1) **Success Rate** – the percentage of episodes in which all task goals were achieved within the step limit (20,000 simulator steps). A task is considered successful only if all

Model	System Feedback	Sim Steps↓	Success Rate↑	Completion Rate↑	Planning Cycles↓	Comm. Rate	Dialogue Length
Llama3.1	W/O Message	3295.20	0.762	0.873	R13.80 / H15.42	-	-
	Opaque	3330.98	**0.787**	**0.890**	R14.83 / H15.41	**65.6%**	2.06±1.35
	Binary	3365.32	0.774	0.875	R14.80 / H15.65	64.8%	2.11±1.37
	Causal	3292.37	0.769	0.880	R22.21 / H15.23	63.1%	2.57±1.62
	Traceable	3360.74	0.775	0.881	R36.94 / H15.03	65.5%	**2.65±1.74**
Llama3.3	W/O Message	3159.14	0.766	0.875	**R12.80 / H13.64**	-	-
	Opaque	3085.18	0.786	0.886	R13.67 / H13.62	56.2%	1.74±1.17
	Binary	**3072.99**	0.776	0.878	R13.67 / H13.63	54.5%	1.82±1.26
	Causal	3112.98	0.785	0.887	R16.13 / H13.84	54.2%	2.06±1.81
	Traceable	3064.87	0.777	0.881	R13.77 / H13.87	55.8%	2.23±1.68

Table 1. Performance on Communicative PARTNR Validation Set. Opaque feedback achieved the highest success rates for both models, despite yielding shorter dialogues than richer modes such as Causal or Traceable. Providing more detailed feedback increased dialogue length but did not improve success, suggesting that excessive information can reduce task efficiency under partial observability. Communication rates remained stable within each model due to identical system prompts, with Llama3.1 generally initiating dialogue more often than Llama3.3.

specified goal conditions are met.

(2) **Completion Rate** – the fraction of sub-goals completed, averaged across episodes. For example, if a task contains four sub-goals and three are completed, the completion rate is 75%. This metric captures partial progress in cases where the full task is not completed.

(3) **Communication Rate** – the proportion of episodes in which the `SendMessage` action was used at least once. This reflects the frequency with which the agents resort to explicit communication.

(4) **Dialogue Length** – the average number of utterances exchanged in episodes where communication occurs, measuring the volume of message exchange. These metrics collectively capture both task performance (success and completion rates) and communication behavior (frequency and length), enabling analysis of how different feedback modes influence coordination effectiveness.

4.2. Overall Evaluation Results

Table 1 summarizes the performance of Llama3.1-70B and Llama3.3-70B agents across the four proposed feedback strategies and the no-communication baseline. Across both models, the Opaque feedback condition consistently achieved the highest task success and completion rates, outperforming richer feedback modes such as Causal and Traceable. For Llama3.3-70B, Opaque reached a success rate of 0.786 and a completion rate of 0.886, representing approximately 2% absolute improvement over the no-message baseline.

Richer feedback modes, particularly Traceable, tended to produce longer dialogues, with the highest average utterance count observed at 2.65 for Llama3.1 and 2.23 for Llama3.3 per communicated episode. Providing partner state information appears to encourage agents to communicate more frequently. However, increased dialogue volume did not correspond to higher task success. Both Causal and Traceable modes underperformed the Opaque baseline in success rate despite showing similar or slightly higher completion rates. This suggests that the current ReAct-style LLM agent does not integrate detailed partner information effectively into task planning and may prioritize message delivery over ongoing task execution.

Communication frequency remained relatively stable within each model across feedback modes, with Llama3.3 maintaining approximately 54–56% of episodes involving communication. This stability is likely due to the identical system prompt used in all experimental conditions, meaning that variations in feedback richness alone did not significantly affect the decision to initiate dialogue. The main difference between models was that Llama3.1 initiated communication in up to 65% of episodes, while Llama3.3 did so roughly 10 percentage points less often, a difference likely arising from inherent reasoning and action-selection tendencies.

The most notable variation across feedback modes was in dialogue length. Minimal-feedback modes such as Opaque and Binary produced shorter conversations, whereas richer modes such as Causal and Traceable resulted in longer and more variable exchanges. Both models showed similar variance patterns, suggesting that detailed feedback systematically increases conversational engagement.

Qualitative inspection further revealed distinct behavioral tendencies. In the Opaque mode, agents tended to focus on task completion and refrained from repeatedly attempting message delivery, instead relying on environmental observations to adjust their plans. By contrast, in the Traceable mode, agents often prioritized ensuring success-

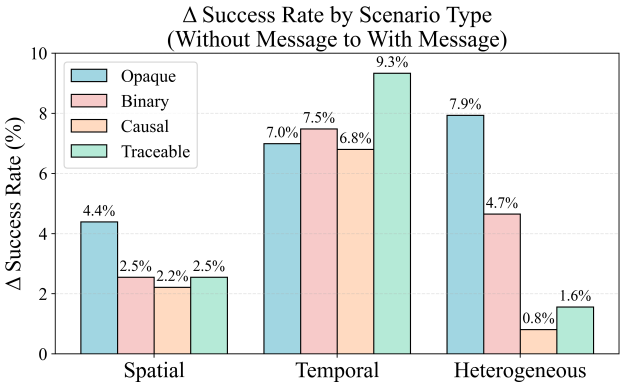


Figure 3. Δ Success Rate across Spatial, Temporal, and Heterogeneous tasks relative to the no-message baseline for each feedback mode. Opaque feedback achieves the largest gains in Spatial and Heterogeneous settings, while Traceable feedback yields the highest improvement in Temporal tasks.

ful message transmission, sometimes abandoning their ongoing task to navigate toward the partner when a delivery failed. This shift in focus appeared to introduce inefficiencies and delayed task execution.

Overall, these results highlight a counterintuitive finding: minimal but well-targeted feedback can yield more robust collaboration under partial observability than verbose, information-rich feedback. We hypothesize that excessive feedback increases the cognitive load on the LLM agent, leading to inefficient communication patterns and execution delays. This points to an important future direction—designing prompt structures and reasoning mechanisms that allow embodied LLM agents to selectively exploit detailed feedback without compromising primary task performance.

4.3. Scenario-wise Analysis

We adopt the four scenario types defined in the PARTNR benchmark to enable a more fine-grained analysis of performance differences across task characteristics. Constraint-free tasks involve simple instructions without ordering or dependency constraints (e.g., "Place a book, a lamp, and a stuffed toy in the cabinet"). Spatial tasks include location or arrangement constraints (e.g., "Place the books on the shelf next to each other"). Temporal tasks require executing sub-goals in a specific order (e.g., "Remove the candles from the table before bringing the plates"). Heterogeneous tasks require combining distinct agent capabilities (e.g., "Move the cushion from the bedroom to the living room sofa, then dust off the cushion" where only human agent can perform the dusting action). In the PARTNR validation set, approximately 25% of episodes are Constraint-free, 35% involve Spatial constraints, 26% involve Temporal ordering, and 22% involve Heterogeneous capabilities. The total exceeds 100% because some tasks exhibit multiple characteristics simultaneously; for example, "First, put the spatula next to the watch on a dining room chair. Then, move them to the counter in the kitchen" combines both spatial arrangement and temporal ordering. Since Constraint-free tasks showed no meaningful improvement with communication, our analysis focuses on the three complex categories-Spatial, Temporal, and Heterogeneous-that better reflect the long-horizon, coordination-intensive scenarios encountered in real-world settings.

In Spatial tasks, Opaque feedback achieved the highest improvement over the no-message baseline, as shown in Figure 3. Agents in the Opaque mode often completed all rearrangements themselves without delegating to the partner. In contrast, other modes exhibited a pattern where an agent that had finished its sub-task sent a completion message to the partner and then terminated. Because message delivery in Opaque mode is uncertain, agents tended to confirm whether the partner was performing the task correctly before ending their own execution. This mutual observation enabled correction and alignment between agents.

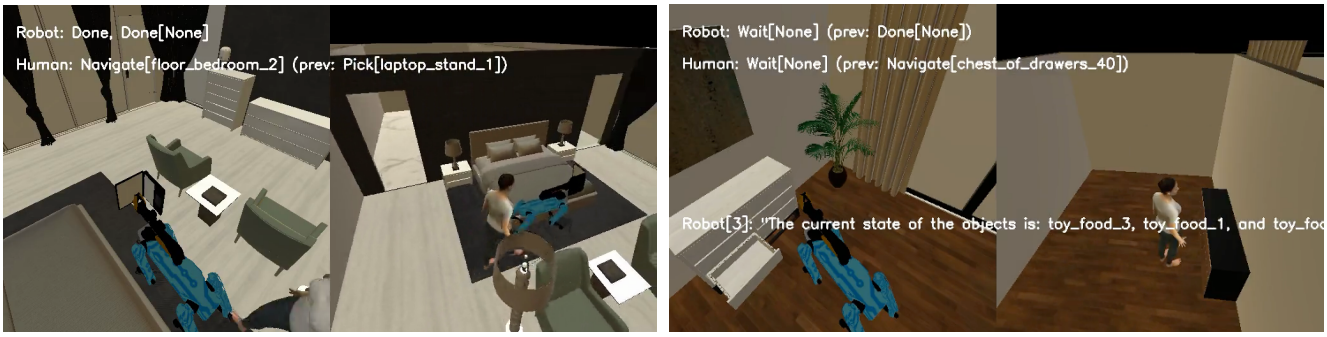
For Temporal tasks, Traceable feedback provided the most notable gains. This mode allowed an agent to infer the partner's ongoing action sequence from spatial cues such as room location and distance, even when a message failed to transmit. Such inference helped maintain correct step ordering in sequential tasks and resulted in more robust execution under partial communication failures. However, we also observed that when messages failed, some agents abandoned their current task and waited to ensure message delivery. This indicates that not only the content of communication but also its timing and prioritization are crucial for maintaining efficiency in temporally constrained scenarios.

In Heterogeneous tasks, Opaque feedback led to the largest improvement, while richer modes such as Causal and Traceable underperformed. These scenarios require agents to identify tasks beyond their capabilities and rely on the partner to execute them. In Causal and Traceable modes, agents frequently focused on exchanging detailed progress information, which sometimes caused confusion in recognizing their own skill limitations. This tendency to prioritize dialogue over execution reduced efficiency, especially when agents were assigned tasks beyond their own capabilities.

4.4. Qualitative Analysis

We qualitatively examine representative episodes to better understand the behavioral differences between agents operating with and without communication. Figure 4 illustrates two such cases.

In the Without Message case, the robot agent completed its part of the task while holding a laptop and terminated prematurely. The human agent, unaware of the robot's ter-



(a) With Out Message Channel

(b) With Message Channel

Figure 4. Qualitative examples from C-PARTNR: (a) The robot prematurely terminates while holding a laptop, causing the human to continue an unproductive search until timeout; (b) The human queries the terminated robot for object locations, receives completion confirmation, and ends the task. Communication enables recovery from misjudged termination conditions, synchronization in sequential tasks, and resolution of deadlock situations.

mination and unable to query its status, continued navigating in search of another laptop until the simulator's step limit was reached, resulting in a failure. In a real-world setting, this situation would require the human to explicitly issue new instructions to the robot. Without a communication channel, agents cannot recover from such misjudgments of termination conditions. This issue was particularly frequent in heterogeneous settings, where the robot misinterpreted its inability to perform an action (e.g., filling a cup with water) as a valid completion and stopped, or entered an infinite loop attempting unsupported actions (e.g., dusting a cushion without a `Clean` skill). Similar deadlock situations were also observed in spatial scenarios, where agents holding objects waited indefinitely for the partner to place down its object before proceeding. While a real human could choose to break the stalemate by placing down their object first, the proxy human agent in simulation exhibited the same blocking behavior as the robot, creating a race condition that could not be resolved without explicit communication.

In the With Message case, the robot agent had completed its task and terminated, but the human agent was unsure whether the overall task was finished. The human initiated a dialogue to request the location of the target objects, to which the robot responded with their placement information, allowing the human to confirm completion and terminate. This example highlights that communication not only facilitates real-time cooperation but also enables agents to verify termination conditions through dialogue. In temporally constrained scenarios, we observed a frequent pattern where an agent would transmit a message upon completing a prerequisite step, enabling the partner to synchronize and proceed with the dependent step. However, in the Binary feedback mode, failed message transmissions often led the agent to repeatedly execute `Wait` actions before retrying the message, whereas for other actions like `Pick`, the agent leveraged system feedback (e.g., "object too far to pick") to plan corrective navigation. These findings underscore that the design of prompt and context handling for communicative agents must account for both the informational content of feedback and its impact on interaction strategies.

5. Conclusion

We explored natural language communication as a means to enhance human–robot collaboration under partial observability in a decentralized setting. Our study with Communicative PARTNR shows that dialogue serves three key purposes. It enables an agent to influence and predict the partner's actions through explicit intention sharing. It resolves conditional deadlocks by allowing timely assistance requests when task progress is blocked by interdependent goals. It facilitates effective sharing of execution history and task-relevant state, preventing misunderstandings and inefficient action loops in temporally constrained tasks.

The results indicate that the effectiveness of communication depends not only on the amount of information exchanged but also on its timing, relevance, and integration into the agent's reasoning. Minimal messages, as observed in the Opaque mode, can yield more robust collaboration than verbose, information-rich exchanges. Future work should focus on developing context engineering methods for embodied LLM agents that can manage conversational content, selectively incorporate partner feedback, and align dialogue strategies with task demands without overloading cognitive resources or disrupting task execution.

Acknowledgment

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (RS-2024-00336738 Development of Complex Task Plan-

ning Technologies for Autonomous Agents 50%, No. RS-2022-II220951 Development of Uncertainty-Aware Agents Learning by Asking Questions 25%) and by the Electronics and Telecommunications Research Institute (ETRI) (24ZB1200 Research of Human-centered Autonomous Intelligence System Original Technology 25%)

References

[1] Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4):819–840, 2002. 2

[2] Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Mottaghi, Priyam Parashar, et al. Partnr: A benchmark for planning and reasoning in embodied multi-agent tasks. ICLR, 2025. 1, 2

[3] Junting Chen, Checheng Yu, Xunzhe Zhou, Tianqi Xu, Yao Mu, Mengkang Hu, Wenqi Shao, Yikai Wang, Guohao Li, and Lin Shao. Emos: Embodiment-aware heterogeneous multi-robot operating system with llm agents. ICLR, 2025. 2

[4] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. ICLR, 2024. 2

[5] Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X Chang, and Manolis Savva. Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation. In *CVPR*, pages 16384–16393, 2024. 3

[6] Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for mind exploration of large language model society. *NeurIPS*, 36:51991–52008, 2023. 2

[7] Xinran Li, Chenjia Bai, Zijian Li, Jiakun Zheng, Ting Xiao, and Jun Zhang. Learn as individuals, evolve as a team: Multi-agent llms adaptation in embodied environments. *arXiv preprint arXiv:2506.07232*, 2025. 3

[8] Jie Liu, Pan Zhou, Yingjun Du, Ah-Hwee Tan, Cees GM Snoek, Jan-Jakob Sonke, and Efstratios Gavves. Capo: Cooperative plan optimization for efficient embodied multi-agent cooperation. ICLR, 2025. 3

[9] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, 2023. Association for Computing Machinery. 2

[10] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander Clegg, Michal Hlavac, So Yeon Min, Vladimír Vondruš, Theophile Gervet, Vincent-Pierre Berges, John M Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars, and robots. CVPR, 2024. 3

[11] SeungWon Seo, SeongRae Noh, Junhyeok Lee, SooBin Lim, Won Hee Lee, and HyeongYeop Kang. Reveca: Adaptive planning and trajectory-based validation in cooperative language agents using information relevance and relative proximity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 23295–23303, 2025. 3

[12] Ryosuke Takata, Atsushi Masumori, and Takashi Ikegami. Spontaneous emergence of agent individuality through social interactions in large language model-based communities. *Entropy*, 26(12), 2024. 1

[13] Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. ICLR, 2024. 2, 4