REVECA: Adaptive Planning and Trajectory-Based Validation in Cooperative Language Agents Using Information Relevance and Relative Proximity

SeungWon Seo*[1], SeongRae Noh*[1], Junhyeok Lee[1], SooBin Lim[1], Won Hee Lee[1], HyeongYeop Kang°[2]

[1]Kyung Hee University, Yongin, Republic of Korea
[2]Korea University, Seongbuk-gu, Republic of Korea
{ssw03270, rhosunr99, bluehyena123, dnpcs, whlee}@khu.ac.kr, {siamiz_hkang}@korea.ac.kr

Abstract

We address the challenge of multi-agent cooperation, where agents achieve a common goal by cooperating with decentralized agents under complex partial observations. Existing cooperative agent systems often struggle with efficiently processing continuously accumulating information, managing globally suboptimal planning due to lack of consideration of collaborators, and addressing false planning caused by environmental changes introduced by other collaborators. To overcome these challenges, we propose the **RE**levance, Proximity, and **V**alidation-**E**nhanced **C**ooperative Language **A**gent (REVECA), a novel cognitive architecture powered by GPT-4o-mini. REVECA enables efficient memory management, optimal planning, and cost-effective prevention of false planning by leveraging Relevance Estimation, Adaptive Planning, and Trajectory-based Validation. Extensive experimental results demonstrate REVECA's superiority over existing methods across various benchmarks, while a user study reveals its potential for achieving trustworthy human-AI cooperation.

Introduction

Digital agents collaborating with humans are crucial in games, educational platforms, and virtual universes. Commonly referred to as Non-Player Characters, these agents play a crucial role in enhancing user immersion in commercial applications, where they assist users in achieving desired objectives. However, these agents operate on prescripted behaviors, limiting their adaptability to complex scenarios and rich conversations with humans. Inspired by the reasoning and communication capabilities of Large Language Models (LLMs) (Li et al. 2023b), we aim to create agents that use LLMs for more effective cooperation in complex environments and improved communication with humans, surpassing traditional methods relying on static, learnable models or reinforcement learning (RL).

This paper introduces REVECA, a **RE**levance, Proximity, and **V**alidation-**E**nhanced **C**ooperative Language **A**gent, an LLM-based agent framework addressing decentralized control, costly communication, complex tasks, partially observable environments, and noisy settings. Similar to prior work (Zhang et al. 2023b), we mainly focus on VirtualHome (Puig et al. 2018), the multi-objective household tasks using well-constructed virtual settings. Specifically, we focus on two decentralized agents cooperating on a multi-objective, long-horizon household task under complex partial observations. Additionally, continuous communication incurs time costs, and irrelevant dummy objects add noise, complicating the environment.

The primary advancements of REVECA over previous works are threefold. Firstly, REVECA significantly reduces both the computational complexity and memory demands of the planning by prioritizing information based on its relevance to the task objectives. This allows the agent to focus on the most pertinent data, thereby maintaining robust performance even in noisy environments typical of real-world scenarios. Information relevance is assessed at the point of acquisition, utilizing the reasoning capabilities of LLMs. Previous research (Zhang et al. 2023b; Li et al. 2023d) stored all scene information in memory, but this led to performance issues due to the fixed context window of LLMs. Some studies (Zhang et al. 2023a, 2024) used queues to retain recent K pieces of information, but this led to suboptimal planning due to limited historical data.

Secondly, REVECA enhances plan optimality by incorporating the relative proximity between collaborators and task objectives. This relative proximity is assessed by inferring potential interaction between collaborators and task objectives, utilizing LLMs and agent's observation data. Previous research on distributed cooperation within partially observed environments (Zhang et al. 2023b,a) has shown that individual agents may be unable to determine whether their optimal plan aligns with the group's best outputs, often leading to suboptimal collective outcomes.

Lastly, REVECA mitigates the occurrence of false plans by implementing a cost-effective plan validation process. In partially observable environments with multiple task objectives, a collaborator's task completion may not be immediately updated in the agent's memory, leading to the creation of redundant plans. To address this, REVECA estimates the likelihood that a collaborator has already completed a given task by inferring potential interaction trajectories. This inference is based on LLMs and historical observation data to predict the collaborators' trajectories, spanning from the last retrieval of the collaborator's information to the current

*These authors contributed equally.
°Corresponding author.

time. Traditional methods (Li et al. 2023c) rely on constant communication for updates, which is costly and prohibitive when cooperating with humans.

To demonstrate REVECA's contributions, this paper presents the results of comparative analysis, ablation studies, and user studies conducted in three multi-room simulation environments: Communicative Watch-And-Help (C-WAH) (Zhang et al. 2023b), ThreeDWorld Multi-Agent Transport (TDW-MAT) (Zhang et al. 2023b), and Noisy-C-WAH. Noisy-C-WAH, a variant of C-WAH with dummy obstacles that are interactive but unrelated to the task objectives, was used to evaluate REVECA's performance in noisy conditions. To further assess the generalization capability of our architecture in a fully observable environment without any modification, the Overcooked-AI (Carroll et al. 2019) environment is also included in the experiments. Results demonstrate that REVECA outperforms recent approaches in terms of success rates, efficiency, and robustness.

Related Work

Cooperative Agents with RL

Research on cooperative agents has a long history (Stone and Veloso 2000; Gronauer and Diepold 2022). The traditional cooperative agent has been studied across various directions, mainly leveraging RL techniques to enable cooperation with diverse collaborators and adapt to dynamic environments (Misra et al. 2018; Amato et al. 2019; Jaderberg et al. 2019; Strouse et al. 2021; Yu et al. 2022; Zhao et al. 2023; Li et al. 2023e, 2024; Zhong et al. 2024). These approaches have facilitated the development of agents that can autonomously learn to cooperate by maximizing cumulative rewards through trial and error in various simulated settings. Notable studies have explored aspects such as mapping state spaces to actions effectively (Misra et al. 2018) and enhancing the robustness of learning algorithms in multi-agent systems (Amato et al. 2019).

To evaluate these approaches, some other researchers have aimed to develop platforms that can test the performance of cooperative agents (Lowe et al. 2017; Savva et al. 2019; Xiang et al. 2020; Puig et al. 2020; Padmakumar et al. 2022; Li et al. 2023a; Zhou et al. 2024). These provide standardized environments to benchmark the performance and generalization capabilities of the agents.

Planning and Decision Making with LLMs

Despite the various studies aimed at developing cooperative agents, a major limitation of previous work has been the lack of natural language-based communication between agents (Das et al. 2018; Carroll et al. 2019; Jaderberg et al. 2019; Puig et al. 2020). Effective language-based communication is crucial for enhancing collaboration, particularly in complex multi-agent environments and when collaborating with humans (Lazaridou, Peysakhovich, and Baroni 2016).

Recently, the advanced reasoning and natural language processing capabilities of LLMs have significantly enhanced agents' decision-making (Li et al. 2022; Wang et al. 2023; Huang et al. 2023; Yuan et al. 2023) and planning abilities (Huang et al. 2022b,a; Li et al. 2023c; Wang et al. 2024).

The integration of LLMs has also significantly improved the development of cooperative LLM-based embodied agents (Zhang et al. 2023b,a; Li et al. 2023d; Zhang et al. 2024). These agents utilize LLMs to understand the environment, plan tasks, and facilitate communication with both human users and other agents. However, existing studies encounter suboptimal performance due to inherent challenges associated with LLMs, such as performance degradation when processing large volumes of input data (Levy, Jacoby, and Goldberg 2024), and inadequate reasoning abilities in handling complex reasoning tasks (Ullman 2023). Furthermore, these studies struggle with the issue of false planning in decentralized and dynamic multi-agent environments. One potential solution involves employing plan evaluation strategies (Madaan et al. 2024; Shinn et al. 2023); however, they have primarily been explored within static, single-agent environments. Another solution is to implement constant communication between collaborators (Li et al. 2023c); however, this incurs substantial communication overhead, making it particularly impractical in scenarios involving human collaborators (Zhang et al. 2023b).

To address the limitations of previous works, we introduce REVECA, an LLM-based cooperative embodied agent framework. REVECA facilitates efficient memory management, optimal planning, and cost-effective prevention of false planning by leveraging information relevance, relative proximity, and plan validation.

Problem Definition

The problem setting of our work is an extension of the decentralized partially observable Markov decision process (DEC-POMDP). Following previous conventions (Bernstein, Zilberstein, and Immerman 2000; Spaan, Gordon, and Vlassis 2006; Zhang et al. 2023b, 2024), our problem is defined as follows. In a state space S, N agents collaborate to achieve a *common goal* $G = \{g_1, \ldots, g_v\}$, which consists of *v* sub-goals. M and A are the memory set and action set of the agent. The memory structure is defined as $M = M_o \cup M_c$, where M_o is *observation memory*, containing object information I_o prioritized based on its relevance to the G, with relevance scores $R = \{Strong, Medium, Low, None\}$. *Strong*, assigned by the agent at observation time, indicates the highest priority. M_c is *collaborator memory*, containing collaborator information I_c. It consists of information about collaborators, including both historical data extracted from the conversation logs and directly observed information in a decentralized, partially observable environment. It encompasses a sequence of communication messages σ_i from the respective collaborator, annotated with the corresponding simulation step i. The Action set $A = A_c \cup A_l$ comprises A_c, which consists of actions related to the transmission of a message σ to a collaborator, and A_l, which consists of pre-defined actions essential for task execution, as specified in the *low-level action skill book*. For simulation steps $i = \{1, \ldots, H\}$, the state transition function T governs the transition from state s_i to state s_{i+1} based on the action a_i, denoted as $T(s_{i+1}, s_i, a) = p(s_{i+1}|s_i, a_i)$. The probability of an agent taking action a_i according to the plan π_i in state s_i is given by $p(a_i|s_i) = p(a_i|\pi_i, s_i)p(\pi_i|s_i, M_i)$. The

simulation runs for a maximum of H steps and will terminate under any of the following conditions: the completion of all sub-goals, reaching the step limit H, or the depletion of viable actions for all agents.

REVECA Framework

Our framework comprises six modules: 1) Communication Module, 2) Observation Module, 3) Memory Module, 4) Planning Module, 5) Validation Module, and 6) Execution Module. These modules incorporate three key processes: Relevance Estimation, Adaptive Planning, and Trajectory-based Validation.

Communication Module facilitates information sharing between agents through natural language, leveraging the advanced capabilities of recent LLMs (Bubeck et al. 2023).

Observation Module is responsible for collecting and categorizing environmental data into four levels of relevance, based on what the agent can observe from its current location. Note that the partial observation environment restricts the agent's observational scope.

Memory Module comprises four components: *common goal*, *observation memory*, *collaborator memory*, and a *low-level action skill book*. This module is responsible for storing, updating, and managing data critical to the agent's decision-making processes.

Planning Module retrieves K information data from the M_o and adaptively generates the plan π, using relevance scores and relative proximity.

Validation Module estimates the likelihood that a collaborator has already completed a given task goal by predicting the collaborator's trajectories. If the plan generated by the Planning Module involves a task goal that has a high probability of being completed by a collaborator, the plan is identified as a false plan, prompting its reformulation.

Execution Module executes validated plans using the pre-defined *low-level action skill book*, which are implemented in Python. The skill book is provided by the benchmark framework.

These modules are invoked throughout REVECA's iterative workflow phases: Observation Time, Planning Time, and Validation Time. Further details on the modular design of REVECA are provided comprehensively in the supplementary materials, and a demonstration is showcased in the accompanying video.

Communication for Information Sharing

The Communication Module, which facilitates information sharing through natural language, is invoked in four cases. First, at the initiation of the simulation, all agents exchange their initial positions and information regarding surrounding objects. Second, when an agent requires another agent's task history for validation purposes. Third, when an agent needs to provide its task history in response to a validation request. Finally, when a sub-goal is completed, the achievement is announced to all other agents.

Observation Time: Relevance Estimation

During the Observation Time, the Observation Module acquires raw scene information and refines it into I_o. I_o encompasses details about objects such as their 3D positions, object IDs and names, room IDs and names, available actions, and object states. In a multi-room environment, the agent's observations are restricted to objects and collaborators within its current room; objects within closed containers (e.g., cabinets or boxes) remain unobserved until accessed. I_o is assigned a R, evaluated by LLMs based on the G. This relevance-based prioritization avoids the reference of all memory entries and simplifies the LLM-based planning process by focusing on the most pertinent information.

I_c is obtained either through direct observation of the collaborator or via natural language communication with other agents. After the communication session terminates, I_c is refined by integrating relevant details extracted from the conversation log using LLMs with information obtained through direct observation. I_c includes the collaborator's held objects, current position, message, and the history of completed plans.

Both I_o and I_c are stored within the Memory Module as M_o and M_c, respectively. An example of how the relevance score is determined and how data is extracted from a communication message is depicted in Figure 1(a).

Planning Time: Adaptive Planning

The Planning Module begins by retrieving K pieces of I_o based on their relevance score and the agent's relative proximity to the associated objects. The relative proximity is calculated as the current distance between the agent, the object position stored in I_o, and the most recent positions of the collaborators stored in I_c. First, all instances of I_o stored in M_o are sorted in descending order according to their relevance score R. For information entries with identical relevance scores, prioritization is further refined by calculating the relative proximity P. Although proximity is naturally a continuous measure, we found that converting numerical distance into corresponding natural language descriptions (e.g. I'm closer than Bob, I'm farther than Bob) significantly enhances LLMs performance. LLMs then utilize zero-shot chain-of-thought prompting (CoT) (Kojima et al. 2022) to generate the plan π by including the retrieved K pieces of I_o, relevance score, and relative proximity in the input prompt, as illustrated in Figure 1(b). To facilitate this process, the agent's current information including the held object, current position, and completed plan history is incorporated to prompt as additional context.

This approach implicitly guides the Planning Module in generating a globally optimal plan among collaborators based on relevance scores and relative proximity.

Validation Time: Trajectory-based Validation

Even a well-constructed plan can become invalid due to environmental changes caused by collaborators during the interval between observation and planning. In partially observable environments, detecting such changes poses a significant challenge for the agent.

A straightforward method to resolve this issue is to revisit the object's location or query all collaborators about their interactions. However, this can lead to inefficient path

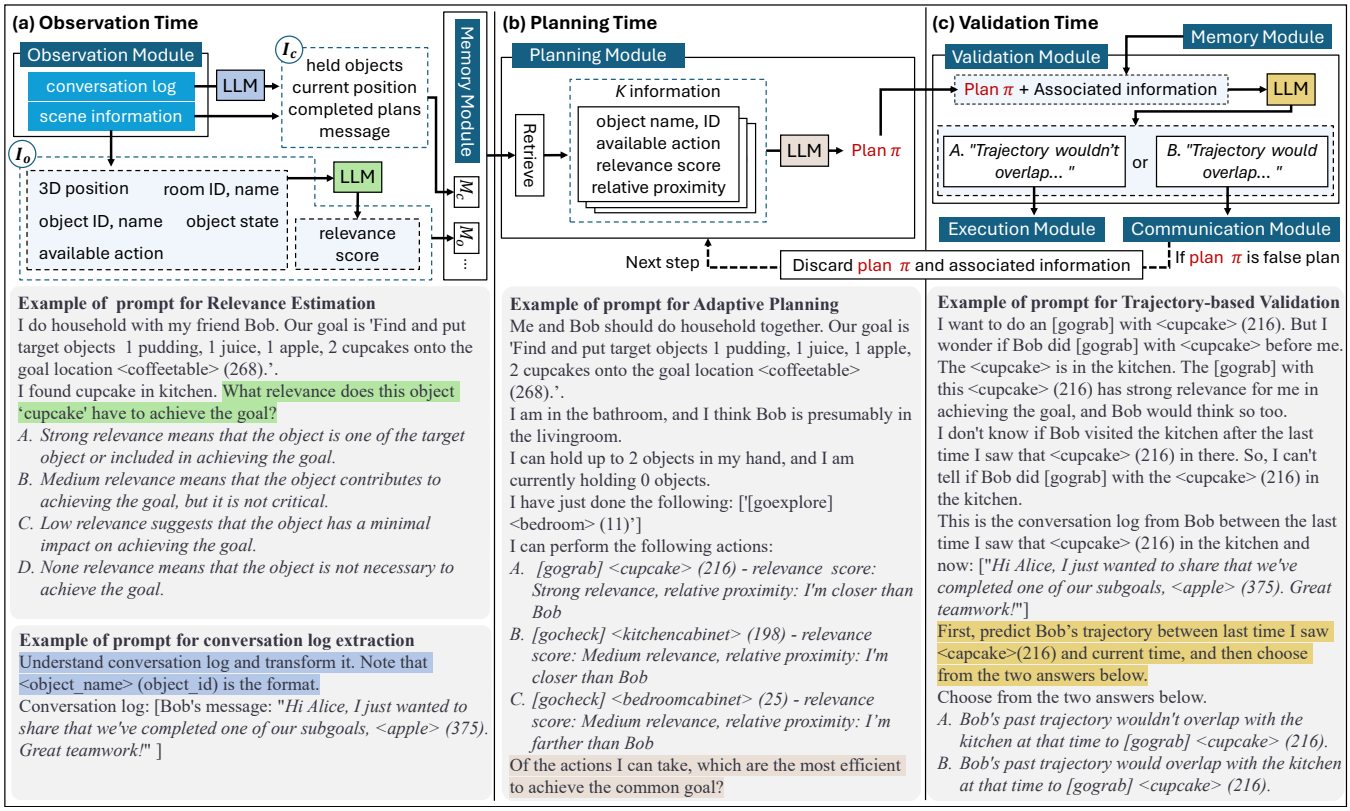

Figure 1: The REVECA process workflow ensures efficient memory management, optimal planning, and cost-effective prevention of false planning through three phases: (a) Observation Time, (b) Planning Time, and (c) Validation Time.

planning and incur substantial communication costs, which is particularly impractical when collaborating with humans.

To address this, REVECA's Validation Module incorporates Trajectory-based Validation, which estimates the validity of a plan using both M_o and M_c. To validate a plan π generated by the Planning Module, the agent predicts each collaborator's past trajectory τ_i, where $1 \leq i \leq N-1$, covering the period from the acquisition time of information used for planning α to the Planning Time β, where $1 \leq \alpha \leq \beta \leq H$. To construct τ_i, all relevant information about collaborator i is retrieved from M_c, along with the relevance scores and I_o from M_o, specifically focusing on the information stored between simulation steps α and β. Given the discontinuity in the associated I_o used in the plan, LLMs' reasoning capabilities are leveraged to infer the missing information, thereby constructing trajectory τ_i.

Based on the plan validity check, if it is determined that no collaborator is likely to have interacted with the object, the agent assumes the plan is valid and proceeds with execution. Otherwise, the agent first sends a message via the Communication Module to the collaborator with the highest interaction probability to confirm the prediction. If the collaborator confirms the interaction, it indicates that the current plan π is a false plan. Consequently, the agent discards both the π and associated I_o used in its formulation and then proceeds with a new planning process during the planning time of the next step. If the collaborator denies the interaction, the agent discards the corresponding τ and queries the next collaborator with the second-highest interaction probability, repeating this process until no potential candidates remain. If all collaborators deny the interaction, indicating current plan π is not a false plan, the agent deems the plan π valid and proceeds with its execution. It is important to note that our environment assumes all agents share a G, possess equal capabilities, and act cooperatively. Therefore, we do not consider scenarios where an agent, despite being fully capable, chooses not to interact with an object necessary for achieving the G. An example of Trajectory-based Validation is depicted in Figure 1(c).

Executing Navigation and Contextual Actions

Once the plan is finalized, the Execution Module retrieves I_o from the M_o to identify the target location. For efficient pathfinding, the A-star search algorithm is employed to navigate the agent toward the object. Upon approaching the object, the agent retrieves an available action from the *low-level action skill book* to execute the planned interaction.

Experiment

We conducted experiments performing multi-objective household tasks using three types of indoor multi-room simulation environments: C-WAH (Zhang et al. 2023b), TDW-MAT (Zhang et al. 2023b), and Noisy-C-WAH—all

of which are partially observable. Additionally, we used the cooperative game simulation, Overcooked-AI (Carroll et al. 2019), which is fully observable, to further evaluate the framework's performance under conditions where complete information is available.

In C-WAH and Noisy-C-WAH, we evaluate the agent performance using Simulation Steps (SS) and Travel Distance (TD) to measure the time cost to achieve the G and the average distance traveled, respectively. We conducted 10 episodes, each with 3 to 5 sub-goals, across two environments, with H set to 250 steps.

In TDW-MAT, performance is evaluated based on the success rate of transporting items, including the overall success rate (TOTAL), and specific success rates for objects categorized as Food (FOOD) and Stuff (STUFF). Each category includes 10 target objects, and H is set to 3000 steps.

In Overcooked-AI, we evaluate the agent performance based on the reward, where a reward of 20 points is obtained each time the two agents successfully complete and serve a dish. We conducted 5 different layouts and H is set to 400 steps. Layout images and detailed descriptions of all experiment settings are provided in the supplementary materials.

REVECA and Baselines

In the comparative experiments conducted in partial observation, our REVECA was evaluated against three baselines: the MCTS-based Hierarchical Planner (MHP), the Rule-based Hierarchical Planner (RHP), and the Cooperative Embodied Language Agent (CoELA). We compared REVECA with MHP and CoELA in C-WAH, with RHP and CoELA in TDW-MAT, and with CoELA in Noisy-C-WAH.

In the comparative experiments conducted in full observation Overcooked-AI, REVECA is evaluated against six baselines: self-play (SP) (Tesauro 1994; Carroll et al. 2019), Population Based Training (PBT) (Jaderberg et al. 2017), Fictitious Co-Play (FCP) (Strouse et al. 2021), Maximum Entropy Population-based training (MEP) (Zhao et al. 2023), Cooperative Open-ended LEarning (COLE) (Li et al. 2023e, 2024), and the ProAgent (Zhang et al. 2023a).

We evaluate the robustness of cooperation between different methods by generating 49 pairs of combinations using our method, REVECA, and the six baselines. The order of the first and second players was reversed for each combination to account for varying starting positions, resulting in the full set of pairs.

To further evaluate the robustness across different versions of LLMs, we conducted experiments using *gpt-4o-mini-2024-07-18* (4o-mini), *gpt-3.5-turbo-0125*, and *Meta-Llama-3.1-8B-Instruct* (Llama 3.1) on the C-WAH and Noisy-C-WAH environments. For all other environments, only 4o-mini was utilized. Additionally, we incorporated GPT-4-driven CoELA performance from the CoELA manuscript (Zhang et al. 2023b) to facilitate a more comprehensive analysis of the capacities of different LLMs. Detailed descriptions of all baseline models and LLMs versions are provided in the supplementary materials to ensure reproducibility.

Method	LLMs	SS ↓	TD (m) ↓
MHP	X	69.40	58.96
CoELA	GPT-3.5	71.90	61.29
CoELA	4o-mini	65.50	53.64
CoELA	GPT-4	57.00	/
REVECA	Llama 3.1	56.00	47.13
REVECA	GPT-3.5	48.90	40.34
REVECA	**4o-mini**	**44.20**	**38.44**

Table 1: Comparative experimental results in C-WAH environment. The best result is highlighted in bold.

Method	LLMs	TOTAL↑	FOOD↑	STUFF↑
RHP	X	0.79	0.83	0.76
CoELA	4o-mini	0.53	0.51	0.55
CoELA	GPT-4	0.71	0.82	0.61
REVECA	**4o-mini**	**0.87**	**0.87**	**0.87**

Table 2: Comparative experimental results in TDW-MAT environment. The best result is highlighted in bold.

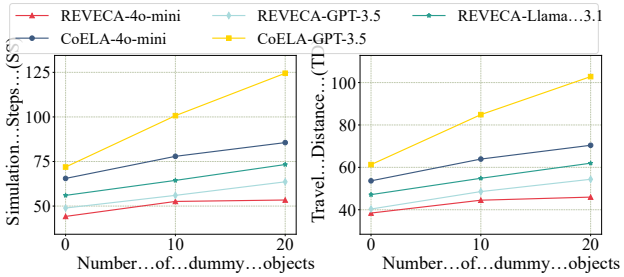

Figure 2: Comparative results in the Noisy-C-WAH environment with varying dummy objects.

Comparative Results: In Partial Observation

Table 1 presents that REVECA outperforms other baseline methods in the C-WAH environment. Notably, the REVECA driven by various LLMs require fewer SS and less TD to complete tasks compared to GPT-4, 4o-mini, GPT-3.5-driven CoELA, and MHP, thereby demonstrating superior efficiency. Interestingly, the performance of GPT-3.5-driven CoELA falls below that of MHP, and GPT-4-driven CoELA underperforms compared to the Llama 3.1-driven REVECA. This demonstrates CoELA's significant dependence on GPT-4's advanced reasoning capabilities.

Table 2 presents results from the TDW-MAT environment, where 4o-mini-driven REVECA outperforms all baselines, including GPT-4-driven CoELA, across all metrics (TOTAL, FOOD, and STUFF).

In the Noisy-C-WAH environment, the experiment included 10 or 20 additional dummy objects. As shown in Figure 2, REVECA models driven by various LLMs outperform CoELA driven by GPT-3.5 and 4o-mini across all metrics. CoELA's strategy of storing all acquired information

in text form leads to a significant decline in reasoning performance, a limitation that becomes even more pronounced when weaker LLMs are used for reasoning. As the number of dummy objects increases, the benefits of utilizing relevance scores become increasingly evident.

Comparative Results: In Full Observation

Table 3 presents Overcooked-AI results, where REVECA, driven by 4o-mini, exhibits high cooperative performance in a fully observable environment, achieving comparable to the state-of-the-art ProAgent, which also utilizes 4o-mini. Notably, our method is not specialized for fully observable environments, demonstrating REVECA's versatility. This finding suggests that REVECA is broadly applicable, not only to household tasks but also to cooperative multi-agent game environments governed by specialized game rules, without requiring any modifications.

As presented in previous research (Zhang et al. 2023a; Carroll et al. 2019), we further conducted comparative experiments in Overcooked-AI by training behavior cloning (BC) models using human data to simulate human users. In this experiment, we tested all pairwise combinations of five BC models and other methods, including REVECA.

As shown in Table 4, REVECA demonstrates superior performance in collaboration with various BC models trained on human data, achieving the highest scores more frequently than ProAgent. Specifically, in the Forced Coordination layout, REVECA consistently achieved high performance regardless of the starting positions. This demonstrates REVECA's robustness in specialized scenarios where each worker is restricted to their designated workspace and cannot substitute for others. This highlights its adaptability and effectiveness in such unique environments.

Ablation Study Results

To demonstrate the significance of each component in our framework, we conducted an ablation study within C-WAH and Noisy-C-WAH environments, the latter augmented with 20 dummy objects. The results are presented in Table 5.

Initially, we tested REVECA in a fully observable setting by enforcing communication before executing any action (full observation). In this setting, each agent is forced to broadcast its perceived information to all other agents before executing any action. While this consumes SS for communication, the agents always maintain up-to-date information and therefore do not generate false plans. One might question whether forced communication is the most convenient approach, but the subsequent user study reveals that this approach is not well-suited for collaboration with humans.

We also tested REVECA under three modified conditions: without using CoT (w/o CoT), without considering relative proximity (w/o proximity), and without using other agents' information (w/o other info). These factors are critical to the reasoning process in REVECA. The experimental results showed a decline in both SS and TD, with the most pronounced reduction occurring when relative proximity was excluded. This finding indicates that the absence of relative proximity impairs the generation of globally efficient paths for the embodied agent, as it fails to consider the collaborator's movements.

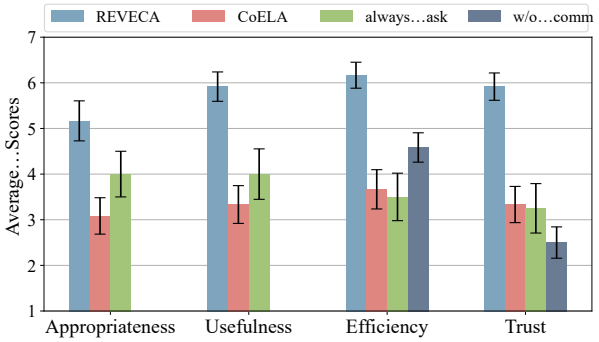

Figure 3: User study results in C-WAH environment. The mean scores and associated standard errors for responses to four research questions.

Next, we evaluated REVECA without using relevance scores, instead employing a distance-based greedy search approach (w/o relevance). While this approach only slightly underperformed compared to REVECA in C-WAH, likely due to the relatively small number of items, it resulted in the lowest performance across all scores in Noisy-C-WAH, except in the K = Inf setting. Additionally, we conducted an experiment by removing the Validation Module (w/o validation), which led to increased SS and TD scores, indicating a performance decline. This decline is attributable to the agent's inability to prevent false plans, thereby hindering the creation of efficient collaborative trajectories.

Lastly, we varied the number of plans K and relevance levels R considered by the LLM planner. The default parameters of K = 3 and R = 4 yielded the optimal performance. In this context, K = 1 corresponds to retrieving only a single piece of information, while K = Inf represents bypassing the retrieval process entirely, thereby imposing the maximum computational load on the LLM planner, as it must directly reference the entire memory.

User Study Results

We conducted a user study to evaluate REVECA's ability to collaborate seamlessly with humans to achieve G. Twelve participants (nine men and three women) with an average age of 23.67 years were recruited. The experiment took place in the C-WAH environment using four methods: REVECA, CoELA, REVECA with an "always ask before action" approach (always ask), and REVECA without communication (w/o comm).

Participants shared the same observation and action space as the agents, interacting with the environment by selecting actions from a predefined list. Each participant completed five sub-goals with each method. After completing each method, they answered a 7-point Likert scale (1: strongly disagree, 7: strongly agree) questionnaire that addressed four key research questions: 1) Did the agent respond appropriately to your intentions? (Appropriateness), 2) Was the interaction with the agent helpful in achieving the goal?

Method	Cramped Room		Asymmetric Advantage		Coordination Ring		Forced Coordination		Counter Circuit	
	Agent1 ↑	Agent2 ↑	Agent1 ↑	Agent2 ↑	Agent1 ↑	Agent2 ↑	Agent1 ↑	Agent2 ↑	Agent1 ↑	Agent2 ↑
SP	165.71	174.29	174.29	197.14	122.86	111.43	34.29	48.57	77.14	68.57
PBT	**182.86**	185.71	197.14	185.71	142.86	142.86	62.86	**91.43**	74.29	57.14
FCP	171.43	**188.57**	177.14	177.14	122.86	142.86	45.71	40.00	57.14	40.00
MEP	180.00	**188.57**	157.14	197.14	174.29	154.29	25.71	42.86	68.57	77.14
COLE	177.14	151.43	205.71	188.57	**177.14**	174.29	37.14	54.29	85.71	108.57
ProAgent	165.71	140.00	260.00	**254.29**	162.86	174.29	85.71	40.00	**125.71**	**131.43**
REVECA	157.14	171.43	**262.86**	234.29	174.29	**177.14**	**88.57**	62.86	120.00	125.71

Table 3: Comparative study results between REVECA and baselines in Overcooked-AI. The best result is highlighted in bold.

Method	Cramped Room		Asymmetric Advantage		Coordination Ring		Forced Coordination		Counter Circuit	
	Agent1 ↑	Agent2 ↑	Agent1 ↑	Agent2 ↑	Agent1 ↑	Agent2 ↑	Agent1 ↑	Agent2 ↑	Agent1 ↑	Agent2 ↑
SP	100.00	80.00	120.00	60.00	48.00	40.00	28.00	12.00	36.00	32.00
PBT	96.00	92.00	124.00	64.00	68.00	64.00	**52.00**	4.00	40.00	28.00
FCP	148.00	148.00	160.00	44.00	112.00	100.00	28.00	32.00	12.00	20.00
MEP	**164.00**	152.00	160.00	64.00	**144.00**	104.00	44.00	32.00	36.00	44.00
COLE	148.00	136.00	164.00	204.00	96.00	92.00	48.00	48.00	90.00	84.00
ProAgent	160.00	**156.00**	212.00	**240.00**	140.00	120.00	24.00	88.00	108.00	**124.00**
REVECA	132.00	140.00	**216.00**	208.00	132.00	**128.00**	**52.00**	**96.00**	**112.00**	112.00

Table 4: Comparative study results with BC Models in Overcooked-AI. The best result is highlighted in bold.

Method	C-WAH		Noisy-C-WAH	
	SS ↓	TD (m) ↓	SS ↓	TD (m) ↓
full observation	45.50	37.26	52.10	42.50
w/o CoT	48.90	43.47	63.90	55.17
w/o proximity	71.60	64.02	74.40	66.18
w/o other info	48.80	42.33	65.00	57.39
w/o relevance	46.20	38.84	82.10	68.52
w/o validation	45.80	40.02	54.60	47.27
$K = 1$	45.20	39.17	57.20	49.01
$K = 2$	45.60	39.91	63.60	53.69
$K = 4$	47.40	40.96	53.50	45.60
$K =$ Inf	53.70	46.45	88.90	77.51
$R = 3$	47.20	40.20	68.30	57.94
$R = 5$	54.90	46.75	76.90	64.86
REVECA	**44.20**	**38.44**	**53.40**	**45.96**

Table 5: Ablation study results in the environments of C-WAH and Nosiy-C-WAH augmented by 20 dummy objects. The best result is highlighted in bold, with the exception of full observation.

(Usefulness), 3) Did the agent's performance help achieve the goal quickly? (Efficiency), and 4) Did you feel a sense of trust with the agent? (Trust) The "w/o comm" method excluded questions on Appropriateness and Usefulness, due to the lack of interaction. Following the questionnaire, participants were interviewed to gather qualitative feedback on each method.

As shown in Figure 3, REVECA scored highest across all four questions, demonstrating its superior performance in human-agent collaboration. Participants noted that CoELA frequently produced messages focused on status reports and planning, rather than directly addressing their questions, which led to lower scores in Appropriateness and Usefulness. In the "always ask" condition, participants found the agent's repetitive questions, which were often of low relevance to their current actions, to be disruptive and demotivating Regarding trust, participants noted that the lack of communication in the "w/o comm" method made it difficult to understand the agent's actions and situation, thereby hindering trust-based collaboration. Further analysis of the user study is provided in the supplementary materials.

Conclusion

In this paper, we introduced REVECA, an LLM-driven cognitive architecture designed for multi-objective household tasks, enabling efficient cooperation between decentralized agents under complex, partially observable environments. By leveraging Relevance Estimation, Adaptive Planning, and Trajectory-based Validation, REVECA enhances agent cooperation in dynamic settings while minimizing communication costs, making it well-suited for human collaboration and effectively managing irrelevant dummy objects. Furthermore, we demonstrate REVECA's generalization capacity in a fully observable game environment. However, REVECA has several limitations. Its effectiveness in open-world outdoor settings with constantly changing remains to be validated. Using *low-level action skill book* could be enhanced by integrating recent advancements in character animation generation technologies. Addressing these limitations could make future versions of REVECA even more robust and applicable across a broader range of multi-agent environments and tasks.

Acknowledgements

This work was supported by ICT Creative Consilience Program through the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT)(IITP-2024-RS-2020-II201819), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2020-II200861), the Ministry of Science and ICT (MSIT), Korea, through the ITRC (Information Technology Research Center) support program (IITP-2024-RS-2024-00438239), the Global AI Frontier Lab project (No. RS-2024-00509257), and the Artificial Intelligence Convergence Innovation Human Resources Development program (No. RS-2022-00155911, Kyung Hee University), all supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP).

References

Amato, C.; Konidaris, G. D.; Kaelbling, L. P.; and How, J. P. 2019. Modeling and Planning with Macro-Actions in Decentralized POMDPs. *The journal of artificial intelligence research*, 64: 817–859.

Bernstein, D. S.; Zilberstein, S.; and Immerman, N. 2000. The Complexity of Decentralized Control of Markov Decision Processes. In *Conference on Uncertainty in Artificial Intelligence*.

Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, J. A.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y. T.; Li, Y.-F.; Lundberg, S. M.; Nori, H.; Palangi, H.; Ribeiro, M. T.; and Zhang, Y. 2023. Sparks of Artificial General Intelligence: Early experiments with GPT-4. *ArXiv*, abs/2303.12712.

Carroll, M.; Shah, R.; Ho, M. K.; Griffiths, T.; Seshia, S.; Abbeel, P.; and Dragan, A. 2019. On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems*, 32.

Das, A.; Gervet, T.; Romoff, J.; Batra, D.; Parikh, D.; Rabbat, M. G.; and Pineau, J. 2018. TarMAC: Targeted Multi-Agent Communication. In *International Conference on Machine Learning*.

Gronauer, S.; and Diepold, K. 2022. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, 55(2): 895–943.

Huang, W.; Abbeel, P.; Pathak, D.; and Mordatch, I. 2022a. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, 9118–9147. PMLR.

Huang, W.; Xia, F.; Shah, D.; Driess, D.; Zeng, A.; Lu, Y.; Florence, P.; Mordatch, I.; Levine, S.; Hausman, K.; et al. 2023. Grounded decoding: Guiding text generation with grounded models for robot control. *arXiv preprint arXiv:2303.00855*.

Huang, W.; Xia, F.; Xiao, T.; Chan, H.; Liang, J.; Florence, P. R.; Zeng, A.; Tompson, J.; Mordatch, I.; Chebotar, Y.; Sermanet, P.; Brown, N.; Jackson, T.; Luu, L.; Levine, S.; Hausman, K.; and Ichter, B. 2022b. Inner Monologue: Embodied Reasoning through Planning with Language Models. In *Conference on Robot Learning*.

Jaderberg, M.; Czarnecki, W. M.; Dunning, I.; Marris, L.; Lever, G.; Castaneda, A. G.; Beattie, C.; Rabinowitz, N. C.; Morcos, A. S.; Ruderman, A.; et al. 2019. Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science*, 364(6443): 859–865.

Jaderberg, M.; Dalibard, V.; Osindero, S.; Czarnecki, W. M.; Donahue, J.; Razavi, A.; Vinyals, O.; Green, T.; Dunning, I.; Simonyan, K.; et al. 2017. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large Language Models are Zero-Shot Reasoners. *ArXiv*, abs/2205.11916.

Lazaridou, A.; Peysakhovich, A.; and Baroni, M. 2016. Multi-agent cooperation and the emergence of (natural) language. *arXiv preprint arXiv:1612.07182*.

Levy, M.; Jacoby, A.; and Goldberg, Y. 2024. Same Task, More Tokens: the Impact of Input Length on the Reasoning Performance of Large Language Models. *ArXiv*, abs/2402.14848.

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

Li, G.; Hammoud, H.; Itani, H.; Khizbullin, D.; and Ghanem, B. 2023b. CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society. In *Neural Information Processing Systems*.

Li, H.; Chong, Y. Q.; Stepputtis, S.; Campbell, J.; Hughes, D.; Lewis, M.; and Sycara, K. P. 2023c. Theory of Mind for Multi-Agent Collaboration via Large Language Models. In *Conference on Empirical Methods in Natural Language Processing*.

Li, S.; Puig, X.; Paxton, C.; Du, Y.; Wang, C.; Fan, L.; Chen, T.; Huang, D.-A.; Akyürek, E.; Anandkumar, A.; et al. 2022. Pre-trained language models for interactive decision-making. *Advances in Neural Information Processing Systems*, 35: 31199–31212.

Li, W.; Qiao, D.; Wang, B.; Wang, X.; Jin, B.; and Zha, H. 2023d. Semantically Aligned Task Decomposition in Multi-Agent Reinforcement Learning. *ArXiv*, abs/2305.10865.

Li, Y.; Zhang, S.; Sun, J.; Du, Y.; Wen, Y.; Wang, X.; and Pan, W. 2023e. Cooperative open-ended learning framework for zero-shot coordination. In *International Conference on Machine Learning*, 20470–20484. PMLR.

Li, Y.; Zhang, S.; Sun, J.; Zhang, W.; Du, Y.; Wen, Y.; Wang, X.; and Pan, W. 2024. Tackling cooperative incompatibility for zero-shot human-ai coordination. *Journal of Artificial Intelligence Research*, 80: 1139–1185.

Lowe, R.; Wu, Y. I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30.

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Misra, D.; Bennett, A.; Blukis, V.; Niklasson, E.; Shatkhin, M.; and Artzi, Y. 2018. Mapping instructions to actions in 3d environments with visual goal prediction. *arXiv preprint arXiv:1809.00786*.

Padmakumar, A.; Thomason, J.; Shrivastava, A.; Lange, P.; Narayan-Chen, A.; Gella, S.; Piramuthu, R.; Tur, G.; and Hakkani-Tur, D. 2022. Teach: Task-driven embodied agents that chat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 2017–2025.

Puig, X.; Ra, K. K.; Boben, M.; Li, J.; Wang, T.; Fidler, S.; and Torralba, A. 2018. VirtualHome: Simulating Household Activities Via Programs. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8494–8502.

Puig, X.; Shu, T.; Li, S.; Wang, Z.; Liao, Y.-H.; Tenenbaum, J. B.; Fidler, S.; and Torralba, A. 2020. Watch-and-help: A challenge for social perception and human-ai collaboration. *arXiv preprint arXiv:2010.09890*.

Savva, M.; Kadian, A.; Maksymets, O.; Zhao, Y.; Wijmans, E.; Jain, B.; Straub, J.; Liu, J.; Koltun, V.; Malik, J.; et al. 2019. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9339–9347.

Shinn, N.; Cassano, F.; Labash, B.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2023. Reflexion: language agents with verbal reinforcement learning. In *Neural Information Processing Systems*.

Spaan, M. T. J.; Gordon, G. J.; and Vlassis, N. A. 2006. Decentralized planning under uncertainty for teams of communicating agents. In *Adaptive Agents and Multi-Agent Systems*.

Stone, P.; and Veloso, M. 2000. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8: 345–383.

Strouse, D.; McKee, K.; Botvinick, M.; Hughes, E.; and Everett, R. 2021. Collaborating with humans without human data. *Advances in Neural Information Processing Systems*, 34: 14502–14515.

Tesauro, G. 1994. TD-Gammon, a self-teaching backgammon program, achieves master-level play. *Neural computation*, 6(2): 215–219.

Ullman, T. D. 2023. Large Language Models Fail on Trivial Alterations to Theory-of-Mind Tasks. *ArXiv*, abs/2302.08399.

Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L.; and Anandkumar, A. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Wang, Z.; Cai, S.; Chen, G.; Liu, A.; Ma, X. S.; and Liang, Y. 2024. Describe, explain, plan and select: interactive planning with LLMs enables open-world multi-task agents. *Advances in Neural Information Processing Systems*, 36.

Xiang, F.; Qin, Y.; Mo, K.; Xia, Y.; Zhu, H.; Liu, F.; Liu, M.; Jiang, H.; Yuan, Y.; Wang, H.; et al. 2020. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11097–11107.

Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; and Wu, Y. 2022. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35: 24611–24624.

Yuan, S.; Chen, J.; Fu, Z.; Ge, X.; Shah, S.; Jankowski, C. R.; Xiao, Y.; and Yang, D. 2023. Distilling script knowledge from large language models for constrained language planning. *arXiv preprint arXiv:2305.05252*.

Zhang, C.; Yang, K.; Hu, S.; Wang, Z.; Li, G.; Sun, Y. E.; Zhang, C.; Zhang, Z.; Liu, A.; Zhu, S.-C.; Chang, X.; Zhang, J.; Yin, F.; Liang, Y.; and Yang, Y. 2023a. ProAgent: Building Proactive Cooperative Agents with Large Language Models. In *AAAI Conference on Artificial Intelligence*.

Zhang, H.; Du, W.; Shan, J.; Zhou, Q.; Du, Y.; Tenenbaum, J. B.; Shu, T.; and Gan, C. 2023b. Building cooperative embodied agents modularly with large language models. *arXiv preprint arXiv:2307.02485*.

Zhang, H.; Wang, Z.; Lyu, Q.; Zhang, Z.; Chen, S.; Shu, T.; Du, Y.; and Gan, C. 2024. COMBO: Compositional World Models for Embodied Multi-Agent Cooperation. *ArXiv*, abs/2404.10775.

Zhao, R.; Song, J.; Yuan, Y.; Hu, H.; Gao, Y.; Wu, Y.; Sun, Z.; and Yang, W. 2023. Maximum entropy population-based training for zero-shot human-ai coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 6145–6153.

Zhong, Y.; Kuba, J. G.; Feng, X.; Hu, S.; Ji, J.; and Yang, Y. 2024. Heterogeneous-agent reinforcement learning. *Journal of Machine Learning Research*, 25: 1–67.

Zhou, Q.; Chen, S.; Wang, Y.; Xu, H.; Du, W.; Zhang, H.; Du, Y.; Tenenbaum, J. B.; and Gan, C. 2024. HAZARD Challenge: Embodied Decision Making in Dynamically Changing Environments. *arXiv preprint arXiv:2401.12975*.