Who is Helping Whom? Analyzing Inter-dependencies to Evaluate Zero-Shot Cooperation in Teams

Anonymous authors
Paper under double-blind review

Abstract

The long-standing research challenge of Zero-shot Cooperation (ZSC) have been tackled by applying cooperative reinforcement learning to train an agent by optimizing the environment reward function and evaluating their performance through task performance metrics such as task reward. However, such evaluation focuses only on task completion, while being agnostic to 'how' the two agents work with each other. Specifically, we are interested in understanding the cooperative behaviors arising within a team - a problem that has been overlooked by the existing literature in MARL. To formally address this problem, we propose the concept of constructive interdependence - measuring how much agents rely on each other's actions to achieve the shared goal - as a key metric for evaluating cooperation in teams. We interpret interdependence in terms of action interactions in a STRIPS formalism, and define metrics that allow us to assess the degree of reliance between the agents' actions. We pair state-of-the-art ZSC agents with other agents for the popular Overcooked domain, and evaluate the task reward and teaming performance for such teams. Our results demonstrate that although trained agents attain high task rewards, they fail to induce cooperative behavior, showing very low levels of interdependence across teams. Furthermore, our analysis reveals that teaming performance is not necessarily correlated with task reward, highlighting that task reward alone cannot reliably measure cooperation arising in a team.

1 Introduction

Achieving zero-shot cooperation (ZSC) to build agents capable of working with a wide range of partners remains a long-withstanding challenge in cooperative RL. This capability is particularly important in the setting of human-agent teaming (HAT) to develop agents that must interact and work alongside humans. Popular approaches in these settings use cooperative reinforcement learning, where agents learn with limited set of training partners and generalize these skills to collaborate with previously unseen partners during deployment. Performance of these agents when paired with a partner are commonly evaluated using metrics such as mean episode rewards over multiple runs (Yu et al., 2023; Strouse et al., 2022; Lou et al., 2023) or the time-steps taken to complete the task in the environment (Sarkar et al., 2023; Zhao et al., 2022a). However, evaluating a team using metrics which measure only the task reward obscures critical details about the performance of the individual teammates and the interactions that arise between them, especially in cases where they can successfully complete the task without necessarily cooperating with each other. Here, we borrow from the distinction introduced in (Zhang et al., 2016), Required cooperation (RC) refers to scenarios where the participation of all team members is necessary to achieve the shared goal whereas non-required cooperation (Non-RC) describes settings where each individual can achieve the goal independently, without relying on the contributions of other teammates. For example, consider a human-agent team working in the environment layout shown in Figure 1 from the Overcooked Game. The players act together in the environment to cook and deliver soups by collecting onions, cooking them in a pot, transferring the soup to a dish, and delivering it at the serving station. The

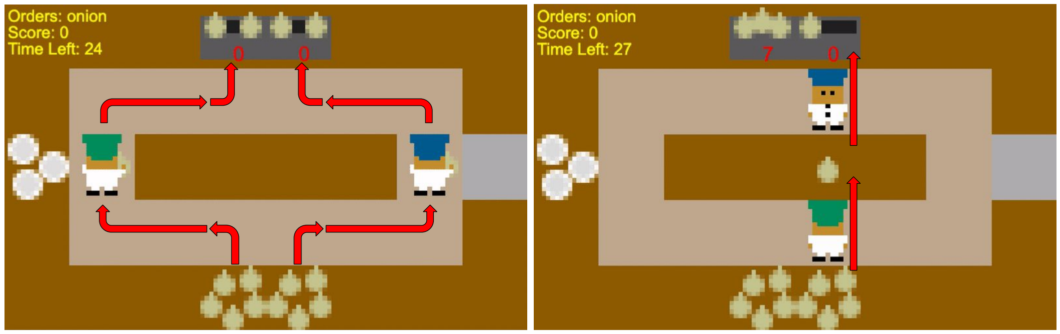


Figure 1: Depicted are two strategies to fill a pot with onions in a cooking game. The coordinated strategy (right) is more efficient than the individual strategies (left), but runs the risk of failure if cooperation is not achieved.

participation of both players is not required to complete the task. The team could complete the task using a strategy with minimal interactions between the teammates, such as one where only the human is completing the task and the agent is merely staying out of the human's way. The team could also be using a strategy where the human and the agent interact and collaborate with each other by using the passing counter to pass onions. Using only the task reward to evaluate the performance would assign the same reward to both teams regardless of their teaming performance. Therefore, in non-RC settings, the task performance is not representative of the teaming performance.

This raises a natural question: *what is the value of cooperation in settings where it is not strictly required to achieve the goal?* To answer this question, it is important to note that ZSC agents are supposed to adapt their behavior and do the best response to diverse partner policies, including those that involve coordination between the teammates. For non-RC settings, Matignon et al. (2012); Fulda & Ventura (2007) describe the *shadowed equilibrium problem* in cooperative RL when there are multiple equilibria (since there are exist multiple way to achieve the task in a non-RC setting, including the agents doing the task independently vs working together). This brings about a persistent challenge in cooperative reinforcement learning is that, during training, agents may fail to encounter cooperative strategies—leading them to converge on behaviors that do not require coordination (Lerer & Peysakhovich, 2019). This creates a significant risk of miscoordination in the ad hoc setting of zero-shot cooperation and human-agent teaming. For instance, in Fig. 1, if one player attempts to follow a cooperative policy—such as passing the onion—but their partner defaults to an uncoordinated strategy, the result is a coordination failure (Carroll et al., 2020). Focusing on only the task reward to evaluate the performance of a ZSC agent hides this fundamental failure of the agent—specifically, the inability of the agent to engage in cooperative behavior when paired with a partner wants to cooperate. Therefore, to truly assess the capabilities of ZSC agents, especially when they are used for human-agent teaming, it is imperative to evaluate their teaming performance—not just their task success. Only by measuring how well agents cooperate with diverse partners can we develop robust, generalizable solutions for real-world collaboration.

In an effort to measure cooperative behavior in a team, we focus on a specific form of cooperation in teams characterized by structured interdependence among team members, as introduced in (Johnson et al., 2020). Such interdependence is central to many real-world teaming applications, as seen for the domains of Urban Search and Rescue (Pateria et al., 2019), collaborative trash removal (Ghavamzadeh et al., 2006), and multi-agent predator-prey systems (Wu et al., 2023; Barton et al., 2018b). Four types of task interdependence have been identified in the study of teamwork: pooled, sequential, reciprocal, and team interdependence (Verhagen et al., 2022; Singh et al., 2016; 2014). In this work, we focus on measuring the sequential and reciprocal interdependencies arising in teams. We propose a novel metric for measuring such interdependencies between multiple agents working as a team, which can be used as a quantifiable measure of cooperation. We map a two-player Markov Game to a symbolic STRIPS formalism, introducing symbolic structure to the world states and the actions, allowing tracking of the interdependencies within the players in a team. We pair state-of-the-art methods trained for zero-shot cooperation for the Overcooked domain with a scripted cooperative agent, human teammates in a user study and in self-play. While our metric is generalizable to any domain, we choose Overcooked because it is a popular benchmark for testing cooperation in multi-agent and human-agent teams, leading to the development of numerous approaches for zero-shot cooperation and human-agent teaming in this domain (Strouse et al., 2022; Carroll et al., 2020; Zhao et al., 2022b; Yu et al., 2023; Li et al., 2024), thus making it a good testbed for assessing the current state-of-the-art. We use the proposed metrics to comprehensively evaluate the teaming performance of teams. Using this metric, we are trying to answer the following research questions - 1. Are trained ZSC agents capable of engaging in cooperative behavior when paired with partners that attempt to initiate cooperation? 2. How does the degree of cooperative behavior vary when these agents are paired with teammates in Required Cooperation (RC) versus Non-Required Cooperation (Non-RC) settings? 3. To what extent do these agents initiate cooperative behavior in teams, and how effectively can they recognize and respond to cooperative intent when it is initiated by their partners? Our results show that ZSC agents are unable to induce/respond to cooperative behavior when paired with partners that attempt to initiate cooperation. Even when the partner follows a known coordination policy, agents don't respond to interdependencies initiated by the partner. In Non-RC settings, teams often achieve high task rewards, but are accompanied by minimal constructive interdependencies, indicating a lack of cooperation arising in these teams. In contrast, in RC settings, higher task rewards are consistently aligned with higher constructive interdependencies. Across all settings, ZSC agents seldom initiate interdependencies themselves and don't respond to those initiated by human teammates. Overall, ZSC agents lack the ability to respond to, or initiate cooperative behaviors in settings where cooperation is helpful but not enforced.

2 Related Works

Previous works in multi-agent teaming use task performance or episodic reward (Strouse et al., 2022; Yu et al., 2023; Zhao et al., 2022b; Li et al., 2024; Wang et al., 2024a; Lou et al., 2023) to evaluate the team's performance. (Zhao et al., 2022a; Knott et al., 2021; Fontaine et al., 2021) emphasize the significance of designing different metrics for evaluation such as collaborative fluency, robot and human idle time etc. (Zhao et al., 2022a) and subjective user studies to measure trust, engagement and fluency of the agents when paired with a human (Zhao et al., 2022a; Ma et al., 2022; Nalepka et al., 2021). However, such metrics depend heavily on specific environment layouts and task structures. Subjective user studies only offer limited insight into the quality of cooperation existing within the team. In contrast, the proposed metric of interdependence is generalizable across domains. Zhang et al. (2024) capture outcomes and certain aspects of the collaboration process such as contribution rate, individual effort, communication frequency whereas Bishop et al. (2020) uses action-based metrics like Productive Chef Actions (PCA), PCA duration, and Chef Role Contribution (CRC) to quantify individual effort and role distribution during task execution. Ries et al. (2024) uses the team member contribution calculated as the difference between the relative proportion of tasks completed by humans versus AI agents. While these metrics measure the contribution of the agents to the task, they do not measure the underlying cooperative dynamics or the structural task dependencies between the actions of the team members. We discuss and compare the interdependence metric with the evaluation presented by Wang et al. (2025), who examine how human-agent teams adapt and evolve over time, focusing on the dynamic processes that shape team interactions and outcomes. Aspects of team formation such as shared goals and team acceptance are measured through subjective perception (Liang et al., 2019), whereas the interdependence metric can objectively reveal whether team members are acting in ways that enable or anticipate each other's contributions. Successful creation and fulfillment of interdependencies indicate role adherence (Wang et al., 2024b), team trust and coordination (Moran et al., 2013; Cai et al., 2019). Johnson et al. (2014; 2020) places interdependence at the center of their model for designing human-machine systems, making it the organizing principle around which the rest of the team's structure and behavior revolves. Barton et al. (2018a); Wu et al. (2023); Barton et al. (2018b) leverage Convergent Cross Mapping (CCM) to measure causal influence between time-series of agent actions, primarily focusing on low-level motion coordination. In contrast, our approach aims to capture more structured and symbolic task interdependencies. Verhagen et al. (2022); Singh et al. (2014) have identified four primary types of task interdependence in teams: pooled, sequential, reciprocal, and team interdependence. Pooled interdependence involves team members working independently without interaction, while sequential interdependence requires tasks to be performed in order. Reciprocal interdependence requires team members taking turns to complete portions of a task, and team interdependence involves concurrent execution of individual tasks with potential joint actions. We define interdependence when the effect of one agent's action satisfies the precondition of another's, modeled through a STRIPS-based formalism. This allows us to identify both unidirectional (sequential) and bidirectional (reciprocal) dependencies.

3 Preliminaries

Two-Player Markov Game : A two-player Markov game for a human-AI cooperation scenario can be defined as $\langle S, A, T, R\rangle$ where S is the set of world states, $A : A_1 \times A_2$ where A_i is set of possible actions for agent i, $T : S \times A_1 \times A_2 \rightarrow S$ is the transition function mapping the present state and the joint action of the agents to the next state of the world, $R_i : S \times A_1 \times A_2 \rightarrow R_i$ is the reward function mapping the state of the world and the joint action to the global reward. For a 2-player cooperative markov game, $R = R_1 = R_2$ where R is the global environment reward function. The joint policy is defined as $\pi = (\pi_1, \pi_2)$ where the policy $\pi_i : S \rightarrow A_i$ is defined for an agent i over set of possible actions A_i. The objective of each agent i is to maximize the expected discounted return $\mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t R(s^t, a_1^t, a_2^t)\right]$ by following the policy π from a given state. Therefore, the policy π is learned by optimizing the task reward received by the agents from the environment.
Multi-Agent Planning : A STRIPS problem is represented as $\langle P, A, I, G\rangle$ where P is the set of propositions which can be used to denote facts about the world, A is the set of planning actions, I is the initial state and G is the goal state. Each fluent $p \in P$ is a symbolic, variable that describes the current state of the environment, with each proposition representing a specific property of an object in the world. The possible fluents for the Overcooked environment can be *counter-empty* - describes whether the counter is empty or not, *pot-ready* - indicates whether the soup is ready in the pot, *soup-served* - indicates whether the soup has been served at the serving station etc. I denotes the propositions representing the initial state of the world and G denotes the propositions corresponding to the goal state of the world. A planning action can be defined as $a =< \text{pre(a)}, \text{add(a)}, \text{del(a)} >$ where pre(a) is the set of propositions that must be true before the action can be executed, add(a) are the propositions that become true after the action is performed and del(a) are the propositions that become false after the action is performed. Extending this to multiple agents, a Multi-agent Planning task can be denoted as $\langle P, N, \{A_i\}_{i=1}^N, I, G\rangle$ where N is the number of agents and A_i is the set of actions for the agent i. We assume that the agents take turns to act and not in parallel. A plan is defined as a sequence of actions $\left(\{a_i^1\}_{i=1}^N, \{a_i^2\}_{i=1}^N, \ldots, \{a_i^n\}_{i=1}^N\right)$ where n is the number of steps in the plan. A plan is a solution Π if it is a sequence of actions that can be applied to the initial state I and results in a world state which satisfies G i.e. $\Pi = \left(\{a_i^1\}_{i=1}^N, \{a_i^2\}_{i=1}^N, \ldots, \{a_i^n\}_{i=1}^N\right)$ is a valid solution plan if $\{a_i^n\}_{i=1}^N\left(\ldots\ldots\left(\{a_i^2\}_{i=1}^N\left(\{a_i^1\}_{i=1}^N(I)\right)\right)\right) \subseteq G$

4 Interdependencies

We pose the teaming problem as a two-player Markov game, where the actions of the teammates take place sequentially. We focus on the case where the team is trying to reach a set of goal states S_G such that $S_G \subseteq S$. The states in S_G are absorbing i.e. $\forall s \in S_G$ and $a_i^G \in A_i$, we have $T(s, \{a_i^G\}_{i=1}^2) = 0$. We represent the solution trajectory for a single agent τ_i as $\tau_i = \left(a_i^t, a_i^{t+1}, \ldots a_i^k \ldots a_i^n\right)$ and the joint-action solution trajectory τ of two agents starting from timestep *t* and reaching a goal state at timestep *n* as $\tau = \left((a_1^t, a_2^t), \left(a_1^{t+1}, a_2^{t+1}\right) \ldots (a_1^n, a_2^n)\right)$. An execution trace *Tr* of a policy π from an initial state s^t as is denoted as $\left(s^t, a^t, s^{t+1}, a^{t+1}, \ldots s^n\right)$, where *Tr* corresponds to the state-action sequence that starts at timestep t and terminates at a goal state $s^n \in S_G$ at a timestep *n*, with $a^k = \left(a_1^k, a_2^k\right)$ and $a_i^k = \pi_i\left(s^k\right)$. The agents receive a task reward R_{task} at the end of *Tr* and τ on reaching the goal state. We define our problem Given the execution trace *Tr* and the joint solution trajectory τ of a team, we only receive R_{task} which does not represent how good or "cooperative" the solution trajectory τ is. To capture the cooperative interactions arising between the teammates in τ, we define the concept of interdependence in the next section.

Mapping the Markov Game to STRIPS : In a Markov Game, the state at a current timestep $s_t \in S$ is typically a high-dimensional vector. s_t can be denoted as a symbolic state with a set of true propositions p_t which denotes the current state of the world. Doing this, we effectively describe each state as a finite set of relevant symbolic facts. Therefore, there exists a function $\mathcal{F} : S \rightarrow P$ mapping the states to symbolic propositions. Here, we refer to Fig. 1. We consider the predicate *counter-empty* to denote if the middle counter is empty. We consider the transition when the green-hat agent (A_2) takes an action to place the onion on the counter. The state at which the agent performs this action has the proposition *counter-empty* set as True, while the action sets *counter-empty* as False in the next state. Therefore, mapping the state to a symbolic state helps us capture the effect of the agents' actions in terms of relevant symbols. We can recall from the execution trace *Tr* of a Markov Game that the state of the world at time *t* is s^t. From s^t, taking action a^t causes the state of the world to change to s^{t+1}. We can map each transition $\left(s^t, s^{t+1}, a^t\right)$ to the symbolic formulation with the help of $\mathcal{F}$. s^{t+1} can be represented as a set of true propositions p_{t+1} and s^t can be represented as p_t. Similarly, we now map the action $a^t = (a_1^t, a_2^t)$ to a symbolic representation. Recall that since the teammates take turns to play, $a^t = (a_1^t, \text{no-op})$ or $a^t = (\text{no-op}, a_2^t)$. For action a_i^t, there exists a mapping from (s^t, a_i^t, s_{t+1}) to a STRIPS style planning action such that $\text{pre}\left(a_i^t\right) \subseteq p_t, \text{add}\left(a_i^t\right) \subseteq p_{t+1}$ and $\text{del}\left(a_i^t\right) \subseteq P \setminus p_{t+1}$. Therefore, the solution trajectory τ can be represented as a joint solution plan Π, where each single-agent action a_i^t in the trajectory can be represented as $a_i^t = \langle \text{pre}(a_i^t), \text{add}(a_i^t), \text{del}(a_i^t) \rangle$. This way we can track the preconditions and effects of the actions of individual agents in the trajectory as symbolic propositions and track the interdependencies between them.

Agent Interdependencies : Given a joint-action solution trajectory τ and the solution trajectory τ_i for an agent i, we define the following properties about τ and τ_i to formalize the concept of interdependence for the solution trajectory:

Definition 4.1. For τ, we define *Interdependence* $\texttt{Int}$ as a pair of actions $(a_i^{t_0+k}, a_j^{t_0})_{i \neq j}$ such that $add(a_j^{t_0}) \subseteq pre(a_i^{t_0+k})$. An *Interdependent* pair of actions $(a_i^{t_0+k}, a_j^{t_0})_{i \neq j}$ has two agents, a *Giver* agent performing the action $a_j^{t_0}$ and a *Receiver* agent performing the action $a_i^{t_0+k}$. Each *interdependent* pair of actions is going to be associated with an object $\texttt{obj}_{int}$.

Definition 4.2. For τ_i, the set of *Trigger* actions is $\texttt{Tr}_i = (a_i \mid \forall a_j \in A_j \cap j \neq i, \text{eff}\,(a_i) \subseteq \text{pre})$ where $a_i \in C_i$.

Definition 4.3. For any object in the world and a starting timestep t_0, the *object influence trajectory* from time t_0, denoted by $\tau_{\texttt{obj}}^{t_0}$, captures all state transitions in the plan from timestep t_0 onward where this object is involved.

$$\tau_{\texttt{obj}}^{t_0} = \{(p_t, a_t, p_{t+1}) \mid t \geq t_0,\ \exists p \in \text{pre}(a_t) \cup \text{add}(a_t) \cup \text{del}(a_t) \text{ where } \texttt{obj} \in O(p)\}$$

where $O(p)$ denotes the set of objects mentioned in proposition p. In other words, $\tau_{\texttt{obj}}^{t_0}$ includes all transitions from timestep t_0 onward where the object explicitly appears in the action's conditions or effects. Also, we have p_n^G i.e. the set of goal predicates at the end of the trajectory. Here, other objects that are affected by *obj* can be captured in the state of that object.

Definition 4.4. An interdependence $\texttt{Int} = (a_i^{t_0+k}, a_j^{t_0})_{i \neq j}$ is a *Goal Reaching Interdependence* if the final state of the object associated with that interdependence (obj_{int}) is also present in the set of goal predicates. Using $p_{\text{final}}^{\texttt{obj}_{\text{int}}}$ to denote the last entry in $\tau_{\texttt{obj}_{\text{int}}}^{t_0}$, I is a goal reaching interdependence if $p_{\text{final}}^{\texttt{obj}_{\text{int}}} \subseteq p_n^G$.

Definition 4.5. Let p_t^{obj} denote the predicate for that object at timestep t, therefore containing nformation about the state of obj at t. An interdependence $\texttt{Int} = (a_i^{t_0+k}, a_j^{t_0})$ associated with object $\texttt{obj}_{\text{int}}$ is said to be a *Non-looping Interdependence* if the following conditions hold; The **giver agent** (agent j), who gives the object $\texttt{obj}_{\text{int}}$ at timestep t_0, does **not** receive the object back in the same state at any future timestep $t > t_0 + k$:

$$\nexists t > t_0 + k, \text{ s.t. agent } j \text{ receives } \texttt{obj}_{\text{int}} \text{ in the same state as at time } t_0$$

. The **receiver agent** (agent i), who receives the object at timestep $k + t_0$, **did not have** the object in that same state at any time $t < t_0 + k$:

$$\nexists t < t_0 + k, \text{ s.t. agent } i \text{ had } \texttt{obj}_{\text{int}} \text{ in the same state}$$

Definition 4.6. An interdependence $\texttt{Int} = (a_i^k, a_j^{k-t})$ is a **Constructive Interdependence**, if it is a *Goal Reaching Interdependence* and a *Non-looping Interdependence*.

Consider a scenario in the Counter Circuit layout where agent j places an onion on the counter at timestep t_0 via action $a_j^{t_0}$, whose effect is $add(a_j^{t_0}) = \{\texttt{onion-on-counter}\}$. Subsequently, at timestep $t_0 + k$, agent i performs action $a_i^{t_0+k}$ to pick up the onion from the counter, with precondition $pre(a_i^{t_0+k}) = \{\texttt{onion-on-counter}\}$. This pair of actions $(a_i^{t_0+k}, a_j^{t_0})$ constitutes a *sequential interdependence* $\texttt{Int}$ linked to the object $\texttt{obj}_{\text{int}} = \texttt{onion}$. The associated object influence trajectory $\tau_{\texttt{onion}}^{t_0}$ captures all state transitions involving the onion, culminating in a final state where the soup contains the onion. Provided that the onion is not returned to agent j in the same state and that agent i had not previously held the onion in that state, this interdependence is *Non-looping*. Consequently, this interaction qualifies as a **Constructive Interdependence**. A *Trigger* action for an agent is placing the onion on the counter, since it could potentially be the precondition for the other agent picking that onion from the counter.

5 Experiment and Results

In this section, we evaluate the performance of state-of-the-art methods (Fig.1 in the Overcooked domain when teamed with a scripted cooperative partner, humans teammates and in self-play, on the forced coordination (RC) and counter circuit (non-RC) layout in Fig. 2. Further details can be found in Section. 7. To assess the quality of cooperation within teams, we use the following metrics: the number of constructive interdependencies ($\texttt{Int}_{\text{cons}}$) and non-constructive interdependencies ($\texttt{Int}_{\text{non-cons}}$). The first metric captures task interdependencies that contribute to task progress and are non-redundant, reflecting efficient and goal-directed coordination. In contrast, the second metric includes interactions that fail to support goal completion or are cyclic, thereby indicating ineffective or unproductive collaboration. In addition, we perform a sub-analysis to measure how many interdependencies are initiated by each team member and how many of those are accepted and acted upon by the teammate.

ZSC Agent paired with Cooperative Partner : We test whether the ZSC agents can successfully adapt to a partner that initiates a coordination policy in a non-RC setting. We consider the coordination policy for the counter circuit layout, as introduced by Carroll et al. (2020) and shown in Fig. 1. In this policy, the green-hat chef puts onions on the counter and the blue-hat chef puts the onions in the pot. Once the soup is ready in the pot, the blue-hat agent picks a dish and serves the soup at the serving station. We set up the game such that a scripted agent performs the role of the green-hat chef, following only their side of the coordination policy and putting onions on the counter. The ZSC agents are assigned the role of the blue-hat chef. This represents a scenario where the agents are paired with a partner who is attempting to perform a known coordination strategy. They are expected to be able to adapt to the actions of the partner and perform actions which complement the green-hat agent's actions, by picking the onion and putting it in the pot. Note that the coordination policy exhibits sequential interdependence, captured by $\texttt{Int}_{\text{cons}}$. Using this metric to assess teams, we can assess the quality of cooperation that emerges when ZSC agents are paired with a scripted cooperative partner. From Table. 1, we observe that all ZSC agents exhibit a low number of interdependencies, despite being paired with a scripted agent that follows a known coordination policy. This suggests that the ZSC agents are largely unable to respond effectively to the partner playing a coordination strategy. Although the scripted partner consistently attempts to initiate interdependencies, most of these efforts are rejected by the ZSC agents. Note that by capturing the sequential interdependencies with a single scalar metric, we avoid the need for analyzing the whole trajectory of the agents, enabling efficient evaluation of key cooperative behaviors. Furthermore, this metric generalizes to any domain with coordination policies exhibiting inter-agent interdependencies, providing a scalable tool for quantifying this kind of cooperation across diverse multi-agent settings.

Agent	Task Reward	Int_{cons}	$Int_{\text{non-cons}}$	$\%P^{\text{trig}}_{\text{tot-sub}}$	$\%P^{\text{not-trig-acc}}_{\text{trig}}$
COLE	36	0.6	1.2	38.5	88.88
MEP	43.33	1.834	1.667	41.28	75.55
HSP	0	0	0.5	28.57	100.0
FCP	0	0	0.167	21.79	100.0

Table 1: ZSC Agents paired with a scripted coordination policy, $\%P^{\text{trig}}_{\text{tot-sub}}$ are the interdependencies that were triggered by the scripted agents, $\%P^{\text{not-trig-acc}}_{\text{trig}}$ are the triggered interdependencies not accepted by the ZSC agents. The task and teaming score are averaged across 20 runs with the scripted agent.

Task vs Teaming Score for ZSC paired with Human Participants : We compute the average task reward and the number of constructive and non-constructive interdependencies for teams of the ZSC agent paired with a human teammate. From Table. 2, we observe that, in Non-RC settings, task reward does not reliably reflect the quality of cooperation. Conversely, in RC settings, there is clear correlation: higher task rewards are consistently accompanied by significantly more constructive interdependencies. We also report that the number of interdependencies in Non-RC domains remains low, highlighting that ZSC agents generally fail to exhibit cooperative behavior when paired with human teammates. From Table. 4a, we observe that human players frequently attempt to initiate cooperative interactions, yet a substantial portion of these are not accepted by ZSC agents. From Table. 4b, even in their highest-scoring runs, ZSC agents in Non-RC settings often achieve task success through independent action, rather than by responding to or building on their human partner's coordination attempts These findings suggest that current ZSC models, while capable of task completion, lack adaptability required for robust human-agent coordination.

Agent	Task Reward		Int_{cons}		$\text{Int}_{\text{non-cons}}$	
	Non-RC	RC	Non-RC	RC	Non-RC	RC
COLE	76.21	56.875	1.89	11.375	3.29	2.875
MEP	50.00	44.102	0.928	8.692	1.285	2.769
HSP	41.11	60.55	1.388	12.055	2.138	3.083
FCP	22.55	35.34	0.97	7.06	0.872	3.441

Table 2: ZSC Agents paired with human teammates; the task reward and teaming metrics are averaged across 36 runs with participants.

Task vs Teaming Performance of ZSC in Self-Play To assess the upper bound of cooperative behavior achievable through self-play, we analyze the top-performing team for each ZSC agent type when paired with an identical copy of itself across both non-RC and RC layouts. This analysis serves to evaluate whether the agents are capable of effective cooperation when paired with itself. Analyzing Table. 3, we observe a consistent pattern across all agent types: constructive interdependence remains low in Non-RC layouts despite agents achieving high task rewards. This indicates a lack of genuine cooperative behavior. In contrast, agents demonstrate markedly higher levels of constructive interdependence in RC settings, aligning more closely with their task performance and suggesting that the dependencies inherent to RC domains facilitate coordination. Crucially, the number of non-constructive interdependencies in Non-RC environments consistently exceeds constructive ones, highlighting that when interdependence does occur, it is often unproductive— looping or irrelevant interactions that do not contribute to task success. These findings further reinforce that task reward alone is not a reliable proxy for cooperative behavior in Non-RC scenarios. Moreover, these findings indicate that even in self-play, ZSC agents fail to induce cooperative strategies.

Agent	Task Reward		Int_{cons}		$\text{Int}_{\text{non-cons}}$	
	Non-RC	RC	Non-RC	RC	Non-RC	RC
COLE	120	200	10.23	30.43	6.58	6.83
MEP	100.00	140	1.05	29.5	10.19	7.53
HSP	120	100	1.82	22.87	12.32	10.667
FCP	60	80	0	16.0	0	16.0

Table 3: Best performing team with ZSC Agents in self-play

Agent	%H-sub$^{\text{trig}}_{\text{tot}-\text{sub}}$		$\%H\text{-}sub^{\text{not trig}-\text{acc}}_{\text{trig}}$	
	Non-RC	RC	Non-RC	RC
COLE	60.28	45.28	70.05	38.34
MEP	66.82	43.57	82.39	39.82
HSP	52.22	42.92	80.85	40.58
FCP	48.30	43.62	98.41	36.84

(a) Analysis of triggered vs accepted interdependencies for the human player.

Agent	%H-sub$^{\text{trig}}_{\text{tot}-\text{sub}}$		$\%H\text{-}sub^{\text{not trig}-\text{acc}}_{\text{trig}}$	
	Non-RC	RC	Non-RC	RC
COLE	40.58	35.89	11.76	5.01
MEP	59.10	46.57	82.39	0.20
HSP	28.67	49.27	33.34	20.58
FCP	44.62	48.68	80.17	18.92

(b) Analysis of triggered vs accepted interdependencies for the best ZSC Agent–human team.

Table 4: Analysis of interdependencies triggered by the human partner vs those accepted by the ZSC agent for human-agent teams, $\%H^{\text{trig}}_{\text{tot-sub}}$ are the interdependencies that were triggered by the scripted agents, $\%H^{\text{not-trig-acc}}_{\text{trig}}$ are the triggered interdependencies not accepted by the ZSC agents.

6 Conclusion

This work evaluates whether Zero-Shot Coordination agents can generalize to behaviors potentially outside their training distribution—particularly when paired with unseen scripted partners or humans who attempt to perform the cooperative policy. To this end, we introduce a metric that captures structured task interdependencies and allows for assessment of cooperation in teams. We also ensure that these interdependencies are constructive—meaning they directly contribute to achieving the team's goal—thereby distinguishing meaningful cooperative interactions from unproductive or redundant ones. Our results show that while ZSC agents achieve high task rewards in non-required cooperation settings, these scores often arise from independent execution rather than actual cooperative behavior. While the agents themselves do not initiate cooperative behavior, they also fail to respond to or build on coordination attempts initiated by partners, including humans, rejecting a majority of triggered interdependencies. In contrast, in Required Cooperation (RC) settings, cooperative behavior—as measured by constructive interdependencies—correlates strongly with task performance. These findings challenge the adequacy of task reward as a standalone metric for evaluating generalizable cooperation in non-RC settings. This work highlights a critical gap in current state-of-the-art for Zero-Shot Coordination: their limited ability to engage in meaningful cooperation when paired with partners attempting to coordinate. Future work would include broadening this metric to include other kinds of interdependencies and cooperative behaviors. Another research direction would be to use the interdependence metric as an additional reward signal to guide learning towards effective cooperation. Prior work by Barton et al. (2018a) has emphasized the importance of explicitly incorporating coordination objectives within learning, rather than relying on coordination to emerge implicitly from the task reward. In non-RC settings, the *shadowed equilibrium problem* (Matignon et al., 2012; Fulda & Ventura, 2007) causes agents to not explore the cooperative strategies during training, since multiple equilibria exist including the non-cooperative strategies. Integrating the interdependence metric as a reward signal could potentially encourage agents to actively recognize and pursue coordination during exploration, potentially learning to play with a diverse set of partners and reducing miscoordination in human-agent teaming scenarios. *Ultimately, this paves the way for developing ZSC agents that not only succeed at tasks but also robustly cooperate with previously unseen partners, thereby enhancing the reliability when deployed in real-world environments.*

References

Sean L. Barton, Nicholas R. Waytowich, and Derrik E. Asher. Coordination-driven learning in multi-agent problem spaces. In *AAAI Fall Symposium: ALEC*, 2018a. URL https://api.semanticscholar.org/CorpusID:52272705.

Sean L. Barton, Nicholas R. Waytowich, Erin Zaroukian, and Derrik E. Asher. Measuring collaborative emergent behavior in multi-agent reinforcement learning, 2018b. URL https://arxiv.org/abs/1807.08663.

Justin Bishop, Jaylen Burgess, Cooper Ramos, Jade Driggs, Tom Williams, Chad Tossell, Elizabeth Phillips, Tyler Shaw, and Ewart de Visser. Chaopt: A testbed for evaluating human-autonomy team collaboration using the video game overcooked!2. pp. 1–6, 04 2020. DOI: 10.1109/SIEDS49339.2020.9106686.

Carrie J. Cai, Emily Reif, Narayan Hegde, Jason Hipp, Been Kim, Daniel Smilkov, Martin Wattenberg, Fernanda Viegas, Greg S. Corrado, Martin C. Stumpe, and Michael Terry. Human-centered tools for coping with imperfect algorithms during medical decision-making, 2019. URL https://arxiv.org/abs/1902.02960.

Micah Carroll, Rohin Shah, Mark K. Ho, Thomas L. Griffiths, Sanjit A. Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-ai coordination, 2020. URL https://arxiv.org/abs/1910.05789.

Matthew C. Fontaine, Ya-Chuan Hsu, Yulun Zhang, Bryon Tjanaka, and Stefanos Nikolaidis. On the importance of environments in human-robot coordination, 2021. URL https://arxiv.org/abs/2106.10853.

Nancy Fulda and Dan Ventura. Predicting and preventing coordination problems in cooperative q-learning systems. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pp. 780–785, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

Mohammad Ghavamzadeh, Sridhar Mahadevan, and Rajbala Makar. Hierarchical multi-agent reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 13(2):197–229, Sep 2006. ISSN 1573-7454. DOI: 10.1007/s10458-006-7035-4. URL https://doi.org/10.1007/s10458-006-7035-4.

Matthew Johnson, Jeffrey M. Bradshaw, Paul J. Feltovich, Catholijn M. Jonker, M. Birna van Riemsdijk, and Maarten Sierhuis. Coactive design: designing support for interdependence in joint activity. *J. Hum.-Robot Interact.*, 3(1):43–69, February 2014. DOI: 10.5898/JHRI.3.1.Johnson. URL https://doi.org/10.5898/JHRI.3.1.Johnson.

Matthew Johnson, Micael Vignatti, and Daniel Duran. *Understanding Human-Machine Teaming through Interdependence Analysis*, pp. 209–233. 08 2020. ISBN 9780429459733. DOI: 10.1201/9780429459733-9.

Paul Knott, Micah Carroll, Sam Devlin, Kamil Ciosek, Katja Hofmann, A. D. Dragan, and Rohin Shah. Evaluating the robustness of collaborative agents, 2021. URL https://arxiv.org/abs/2101.05507.

Adam Lerer and Alexander Peysakhovich. Learning existing social conventions via observationally augmented self-play, 2019. URL https://arxiv.org/abs/1806.10071.

Yang Li, Shao Zhang, Jichen Sun, Wenhao Zhang, Yali Du, Ying Wen, Xinbing Wang, and Wei Pan. Tackling cooperative incompatibility for zero-shot human-ai coordination, 2023.

Yang Li, Shao Zhang, Jichen Sun, Wenhao Zhang, Yali Du, Ying Wen, Xinbing Wang, and Wei Pan. Tackling cooperative incompatibility for zero-shot human-ai coordination, 2024. URL https://arxiv.org/abs/2306.03034.

Claire Liang, Julia Proft, Erik Andersen, and Ross A. Knepper. Implicit communication of actionable information in human-ai teams. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pp. 1–13, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359702. DOI: 10.1145/3290605.3300325. URL https://doi.org/10.1145/3290605.3300325.

Xingzhou Lou, Jiaxian Guo, Junge Zhang, Jun Wang, Kaiqi Huang, and Yali Du. Pecan: Leveraging policy ensemble for context-aware zero-shot human-ai coordination, 2023. URL https://arxiv.org/abs/2301.06387.

Lanssie Mingyue Ma, Martijn Ijtsma, Karen M. Feigh, and Amy R. Pritchett. Metrics for human-robot team design: A teamwork perspective on evaluation of human-robot teams. *J. Hum.-Robot Interact.*, 11(3), September 2022. DOI: 10.1145/3522581. URL https://doi.org/10.1145/3522581.

Laetitia Matignon, Guillaume J. Laurent, and Nadine Le Fort-Piat. Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. *The Knowledge Engineering Review*, 27(1):1–31, 2012. DOI: 10.1017/S0269888912000057.

Stuart Moran, Nadia Pantidi, Khaled Bachour, Joel E. Fischer, Martin Flintham, Tom Rodden, Simon Evans, and Simon Johnson. Team reactions to voiced agent instructions in a pervasive game. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, IUI '13, pp. 371–382, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450319652. DOI: 10.1145/2449396.2449445. URL https://doi.org/10.1145/2449396.2449445.

Patrick Nalepka, Jordan Gregory-Dunsmore, James Simpson, Gaurav Patil, and Michael Richardson. Interaction flexibility in artificial agents teaming with humans. 07 2021.

Shubham Pateria, Budhitama Subagdja, and Ah-Hwee Tan. Multi-agent reinforcement learning in spatial domain tasks using inter subtask empowerment rewards. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 86–93, 2019. DOI: 10.1109/SSCI44817.2019.9002777.

Anthony J. Ries, St'ephane Aroca-Ouellette, Alessandro Roncone, and Ewart de Visser. Gaze-informed signatures of trust and collaboration in human-autonomy teams. *ArXiv*, abs/2409.19139, 2024. URL https://api.semanticscholar.org/CorpusID:272987912.

Bidipta Sarkar, Aditi Talati, Andy Shih, and Dorsa Sadigh. Pantheonrl: A marl library for dynamic training interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 13221–13223, 2022.

Bidipta Sarkar, Andy Shih, and Dorsa Sadigh. Diverse conventions for human-ai collaboration, 2023. URL https://arxiv.org/abs/2310.15414.

Ronal Singh, Tim Miller, and Liz Sonenberg. A preliminary analysis of interdependence in multiagent systems. volume 8861, 12 2014. ISBN 978-3-319-13190-0. DOI: 10.1007/978-3-319-13191-7_31.

Ronal Singh, Tim Miller, and Liz Sonenberg. Communication and shared mental models for teams performing interdependent tasks. 05 2016. ISBN 9783319665948. DOI: 10.1007/978-3-319-46882-2_10.

DJ Strouse, Kevin R. McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data, 2022. URL https://arxiv.org/abs/2110.08176.

R. Verhagen, Mark Antonius Neerincx, and M. Tielman. The influence of interdependence and a transparent or explainable communication style on human-robot teamwork. *Frontiers in Robotics and AI*, 2022.

Mengyao Wang, Jiayun Wu, Shuai Ma, Nuo Li, Peng Zhang, Ning Gu, and Tun Lu. Adaptive human-agent teaming: A review of empirical studies from the process dynamics perspective. 2025. URL https://api.semanticscholar.org/CorpusID:277787105.

Xihuai Wang, Shao Zhang, Wenhao Zhang, Wentao Dong, Jingxiao Chen, Ying Wen, and Weinan Zhang. ZSC-eval: An evaluation toolkit and benchmark for multi-agent zero-shot coordination. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024a. URL https://openreview.net/forum?id=9aXjIBLwKc.

Zekun Moore Wang, Zhongyuan Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, Man Zhang, Zhaoxiang Zhang, Wanli Ouyang, Ke Xu, Stephen W. Huang, Jie Fu, and Junran Peng. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models, 2024b. URL https://arxiv.org/abs/2310.00746.

Haochen Wu, Amin Ghadami, Alparslan Emrah Bayrak, Jonathon M. Smereka, and Bogdan I. Epureanu. Evaluating emergent coordination in multi-agent task allocation through causal inference and sub-team identification. *IEEE Robotics and Automation Letters*, 8(2):728–735, 2023. DOI: 10.1109/LRA.2022.3231497.

Chao Yu, Jiaxuan Gao, Weilin Liu, Botian Xu, Hao Tang, Jiaqi Yang, Yu Wang, and Yi Wu. Learning zero-shot cooperation with humans, assuming humans are biased, 2023. URL https://arxiv.org/abs/2302.01605.

Shao Zhang, Xihuai Wang, Wenhao Zhang, Yongshan Chen, Landi Gao, Dakuo Wang, Weinan Zhang, Xinbing Wang, and Ying Wen. Mutual theory of mind in human-ai collaboration: An empirical study with llm-driven ai agents in a real-time shared workspace task, 2024. URL https://arxiv.org/abs/2409.08811.

Yu Zhang, Sarath Sreedharan, and Subbarao Kambhampati. A formal analysis of required cooperation in multi-agent planning. In *Proceedings of the Twenty-Sixth International Conference on International Conference on Automated Planning and Scheduling*, ICAPS'16, pp. 335–343. AAAI Press, 2016. ISBN 1577357574.

Michelle Zhao, Reid Simmons, and Henny Admoni. Coordination with humans via strategy matching, 2022a. URL https://arxiv.org/abs/2210.15099.

Rui Zhao, Jinming Song, Yufeng Yuan, Hu Haifeng, Yang Gao, Yi Wu, Zhongqian Sun, and Yang Wei. Maximum entropy population-based training for zero-shot human-ai coordination, 2022b. URL https://arxiv.org/abs/2112.11701.

Supplementary Materials

The following content was not necessarily subject to peer review.

7 Environment Details

The team of 2 players is in a gridworld environment with onion dispensers, dish dispensers, pots, serving stations, and empty counters. The players can either move in the environment or interact with these objects. The objective of the game is to cook and deliver three soups as quickly as possible. To do this, the team must do the following tasks: pick and drop three onions from the onion dispenser, place them in the cooking pot, and wait for the soup to be done. The next steps are to pick a dish from the dish dispenser, transfer the cooked soup to the empty dish, and deliver the soup to the serving station. Each player and each counter can hold only one object at a time. On successful delivery of a soup, both the players receive the task reward. Therefore, both players are incentivised to collaborate to prepare the soup and deliver it as many times as possible. The environment is fully observable and communication is not allowed between agents in the environment.

SOTA Methods: FCP Strouse et al. (2022), MEP Zhao et al. (2022b),HSP Yu et al. (2023) and COLE Li et al. (2024) are trained using a two-stage training framework, where a diverse partner population is created through self-play in the first stage, followed by the second stage where the ego agent is iteratively trained by having it play against sampled partners from the population and optimizing mainly the task reward using reinforcement learning. All these methods focus on improving the diversity of the partner population in the first stage. While MEP adds maximum entropy to the reward for increasing the diversity of the population, HSP tries to model the human teammate's reward as event-based rewards to construct a set of behavior-preferring agents. COLE presents cooperative games as graphic-form games and calculates the reward from the cooperative incompability distribution. The ego agent in all these approaches are trained to optimize the episodic task reward, which is also the objective metric being used to measure cooperation when these agents are paired with an unseen teammate (also human). We recruited 36 participants from our university in the range from 18 to 31 who were pursuing either an undergraduate or a graduate degree. We initially conducted a pilot study on 5 participants spread across each of the two evaluation domains. The final study, refined using the pilot study responses, had a sample size of 31 participants. Participants had an average age of 20.75 years, and a median age of 22.5 years. Out of the 36, there were 24 male participants and 12 female participants. 23 participants (63.9%) reported to not have any familiarity with playing the Overcooked game earlier, and the remaining 13 (36.1%) were familiar with the game.

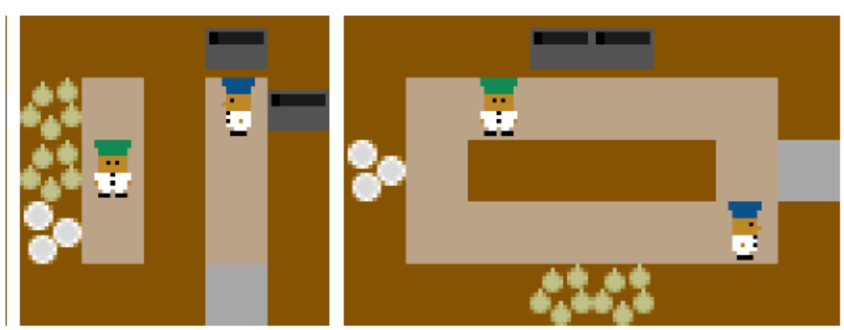


Figure 2: Left: Forced coordination layout which is a required cooperation (RC) setting. Right: Counter circuit layout which is an non-required cooperation (Non-RC) setting.

8 Pipeline

To support reproducibility and generalizability of our proposed cooperation metric, we provide a domain-agnostic software package[1] that allows researchers to apply our analysis across any multi-agent domains. While the main paper demonstrates the utility of the metric in the Overcooked environment, the framework is explicitly designed to be decoupled from any domain-specific assumptions. The system is structured into two independent modules (which are described in detail in the next two sub-sections):

(1) Mapping Module: This module abstracts execution traces into a symbolic representation, generating the grounded trajectory. Given a trajectory $\tau = (s^t, a^t, s^{t+1})_{t=0}^{T}$ from any Markov Game, the module uses a user-defined mapping function $\mathcal{F} : S \to 2^P$ to convert each low-level state s^t into a set of true symbolic propositions $p_t \subseteq P$, where P is the set of domain predicates. Likewise, each agent action a_i^t is mapped into a STRIPS-style operator $\langle \text{pre}(a_i^t), \text{add}(a_i^t), \text{del}(a_i^t) \rangle$, derived from the symbolic state transitions $(p_t, pt + 1)$. The mapping configuration—defining predicates, object types, and effect extraction functions—is modular and can be specified declaratively for any domain.
(2) Analysis Module: This module performs an interdependence analysis on the grounded trajectory by examining how the effects of one agent's action satisfy the preconditions of subsequent actions by teammates. The analysis module classifies such interactions into constructive (task-contributing) and non-constructive (redundant or not task-contributing) interdependencies. This module generates the count of each type of interdependence in the team's action trajectory in one round of the game.

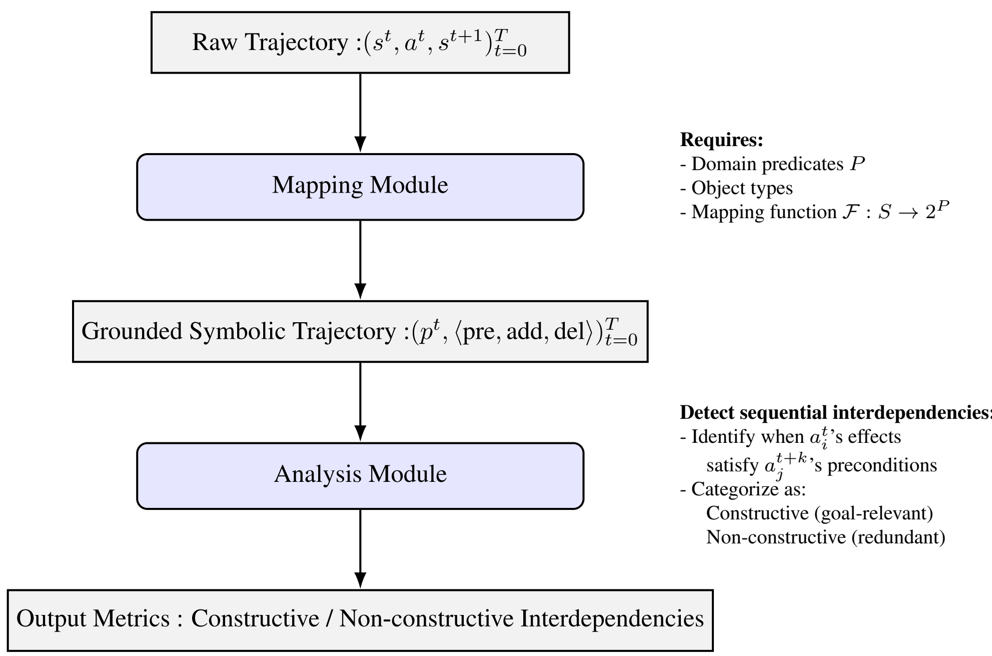


Figure 3: Software architecture for our domain-agnostic cooperation analysis framework. The Mapping Module converts raw trajectories to symbolic STRIPS-style traces, and the Analysis Module identifies interdependencies

8.1 Mapping Module

The `mapping` module provides a general-purpose utility to convert trajectories from any Markov Game environment into a symbolic STRIPS-like planning formalism expressed in PDDL. This

[1]Repository: https://anonymous.4open.science/r/neu25/

abstraction is achieved by defining a declarative mapping between environment states and a set of domain-specific predicates that describe the symbolic state of the world.

The module is designed to be domain-agnostic. Users define a configuration file specifying:

- The list of symbolic predicates relevant to their environment.
- Custom extraction functions for identifying which predicates hold in a given state.
- Mappings from low-level environment actions to high-level symbolic actions, including their preconditions, add effects, and delete effects.

Given a trajectory consisting of (s^t, a^t, s^{t+1}) tuples, the mapping module automatically generates:

- A symbolic trace of world states $p_t = \mathcal{F}(s^t)$.
- A sequence of STRIPS-style operator instances for each agent's action, of the form:

$$a_i^t = \langle \text{pre}(a_i^t), \text{add}(a_i^t), \text{del}(a_i^t) \rangle.$$

The output is a valid, grounded PDDL trace. Internally, the codebase is modular and allows plugging in new domain environments with minimal changes — only the symbolic interface for states and actions needs to be defined. This module supports multi-agent turn-based trajectories by assuming alternating agent moves and handles each agent's action separately when computing symbolic transitions. Conflicts arising from simultaneous execution are handled in the mapping module, so although each agent's moves are processed independently, the code remains fully generalizable to any multi-agent environment.

Algorithm 1 Convert Grounded Trajectory to PDDL Trace Logs (`convert_traj_to_pddl`)

Require: `trajectory`: list of timesteps, each containing a list of (agent, action) pairs
Ensure: $(\mathit{Grids}, \mathit{Logs})$: sequence of grid states and action-logs per timestep

```
grid ← InitGrid()
Grids ← [];    Logs ← []
for each timestep t = 0 to |trajectory| − 1 do
    stepActions ← trajectory[t].action
    logCurrent ← {}
    for each (agent, act) ∈ stepActions do
        (pre, eff, del, grid) ← ApplyAction(act, grid, agent)
        logCurrent[agent] ← {pre_conditions : pre, effects : eff, deletes : del}
    end for
    Append (clone(grid)) to Grids
    Append logCurrent to Logs
end for
return (Grids, Logs)
```

Key Helper Functions:

- `ApplyAction(action, grid, agent_index)`: Applies the specified action for the given agent on the current grid state, returning the pre-conditions, effects, deletes list, and the updated grid. **Note:** This function is domain-dependent and must be implemented according to the specific dynamics and action schema of your environment.

8.2 Analysis Module

The analysis module, as depicted in Algorithm 1, provides a domain-agnostic framework for detecting and categorizing interdependent interactions between agents within a multi-agent environment. Given a sequence of environment states (`snapshots`) and corresponding action logs parsed from PDDL traces (generated by the mapping module, the algorithm dynamically maintains effect lists for each agent. At each timestep, the algorithm systematically checks whether the preconditions of an agent's action are satisfied by the effects of another agent's prior actions, thereby identifying potential interdependencies. Each detected interdependence is further classified into constructive, looping, irrelevant, or non-constructive categories by evaluating whether the object involved contributes to a goal, is repeatedly exchanged, or is otherwise extraneous. This modular design enables the analysis code to be readily applied across different domains, provided that the environment logs have been mapped to a consistent PDDL schema by the mapping module.

Algorithm 2 Detecting Interdependencies and Their Types in the grounded state and action trajectory (`detect_int`)

Require: Data logs: `snapshots` (state log), `action_logs`;
Ensure: Counts of interdependencies along with their types, and lists of actions by each agent which triggered an interdependence.

For each agent: effect_list[agent] ← [ ] ▷ Initialize empty effect list
for each timestep t up to trajectory length **do**
  for each agent **do**
    if agent delivers an object **then**
      Record the delivered object in goal objects array
    end if
  end for
end for
for each timestep t up to trajectory length **do**
  for each agent **do**
    effect_list[agent] ← `filter_effect_list_by_state`(effect_list[agent], snapshots[t])
    Check if the current action's precondition matches an effect in the other agent's effect list via `check_precondition_in_effect_list`
    if precondition matches **then**
      Assess:

      Goal-reaching: Is the object part of the goal? (`check_if_int_goal`)

      Giver loop: Does the object return to the giver in the same state? (`check_if_giver_loop`)

      Receiver loop: Did the receiver ever possess the object in the same state? (`check_if_receiver_loop`)

      if all conditions met **then**
        Increment constructive interdependencies count
      else if loops detected **then**
        Increment looping interdependencies count
      else if not goal-reaching **then**
        Increment irrelevant interdependencies count
      else
        Increment non-constructive interdependencies count
      end if
    end if
  end for
  Save deep copy of current effect lists for next timestep
end for
return Interdependence counts of four types, list of trigger actions for each agent

Key Helper Functions:

- `extract_cells_with_object(grid_state)`: Extracts cells containing an 'object' property.
- `filter_effect_list_by_state(effect_list, state_snapshot)`: Filters and deduplicates effect entries by verifying object presence and state against the snapshot.
- `check_precondition_in_effect_list(action, effect_list_other_agent)`: Checks if an action's precondition matches any effect in another agent's effect list.
- `check_if_int_goal(int_obj_id, goal_object_arr)`: Determines if an object is part of the goal.
- `check_if_giver_loop(int_obj_id, giver_agent_id, snapshots)`: Checks if a giver receives the object back.
- `check_if_receiver_loop(int_obj_id, rec_agent_id, snapshots)`: Checks if the receiver already held the object.

9 Illustrating Evaluation of Cooperative Behavior in a Search and Rescue Domain

We demonstrate that the proposed metric for measuring cooperation generalizes naturally to a heterogeneous Search and Rescue (SAR) domain. The domain simulates a common emergency setting—a house partially engulfed in flames with multiple victims scattered throughout. The scenario is modeled on a discrete 2D grid representing rooms and hallways within the house. Some areas are blocked by debris or actively burning fires, and victims may be located in proximity to these hazards. Successful rescue requires coordinated efforts from a heterogeneous team of agents — each with specialized capabilities and constraints. With its heterogeneous team of firefighters and nurses, this domain provides a rich testbed for analyzing cooperative behavior.

9.1 Domain Specification

We define the SAR environment as:

$$\mathcal{G}_{\text{SAR}} = \langle \mathcal{I}, \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$$

- *Agents:* $\mathcal{I} = \{\text{Nurse (N)}, \text{Firefighter (F)}\}$
  - **Nurse (N)**: Can treat victims without a medical kit as well as administer aid using a medical kit to victims.
  - **Firefighter (F)**: Can extinguish fire using a fire extinguisher.

  The locations and states of all the victims is unknown to the agents upon initialization. All the agents explore the space to discover new victims.
- *State Space:* $\mathcal{S}$ includes:
  - **Agent Locations:** The grid coordinates of each agent.
  - **Victim Locations:** The positions of all victims in need of rescue.
  - **Victim Status:** Each victim may be in one of two states: `untreated` or `treated`.
  - **Cell Conditions:** Each grid cell can contain:
    - `Debris` (present or cleared),
    - `Fire` (burning or extinguished).
  - **Agent Inventories:** For each agent, a list of carried objects (e.g., medical kit, fire extinguisher).
  - **Guard Status:** A Boolean flag indicating whether an agent is currently being guarded by a police agent.

- *Actions:* Each agent has a discrete action space consisting of five actions: —up, down, left, and right and an interact action that allows it to engage with objects in the environment.
- *Transition Function T:* The environment transitions are governed by object-agent interactions and spatial constraints. The transition function $T(s, a, s')$ depends on the current state s, the agent's action a, and environmental conditions. Some examples of critical transition functions in this domain are:
  - **Blocked Movement:** Movement actions are invalid or fail if the target cell contains uncleared debris or active fire.
  - **Interact(Firefighter, Extinguisher,Fire:** Fire in the target cell is extinguished.
  - **Interact(Nurse,Medical Kit,Victim):** Victim status transitions from `untreated` to `treated` within 20 timesteps. It takes 100 timesteps if there is a fire in the room.
  - **Interact(Firefighter,Medical Kit,Nurse:)** Transfers medical kit from firefighter to nurse.
  - **Interact(Firefighter,Debris, Cell):** Clears debris in the current cell.
- *Reward Function R:* At the end of a run of a fixed number of timesteps, all agents receive +10 for each victim successfully treated.

9.2 Mapping to PDDL

The Search and Rescue (SAR) domain described above can be seamlessly integrated with the `mapping` module to produce grounded symbolic trajectories. By specifying a domain configuration file, users can declaratively define the set of symbolic predicates (e.g., `VictimLocationKnown`, `Has(Nurse, MedicalKit)`, `FireExtinguished`), along with extraction functions that detect these predicates from environment states. Low-level actions, such as `Interact(Nurse, MedicalKit, Victim)`, are mapped to high-level symbolic operators with well-defined preconditions and effects. As agents traverse the environment and execute actions, the `mapping` module produces a symbolic trace that reflects the evolving state of the environment and the effects of agent actions, in a post-hoc manner.

9.3 Interdependencies in the SAR Domain

Once trajectories are converted into grounded symbolic traces by the mapping module, the `analysis` module can be directly applied to detect and categorize interdependent interactions among agents. The analysis algorithm, as described in Algorithm 1, processes these traces to dynamically track how agent actions influence one another. We can now formally define interdependencies between agents in the SAR domain. We illustrate examples of sequential interdependencies below:

Example 1: Firefighter discovers victim $\rightarrow$ Nurse treats victim : In this domain, firefighter and nurse agents collaboratively explore the environment to locate and assist victims. While they may search independently to maximize spatial coverage, coordination enables them to operate in parallel effectively. In this example, Firefighter 1 (F1) discovers Victim 1 (V1) by performing the action $a_j^{t_0}$ = `Interact(Firefighter, Victim)`, which results in the predicate `VictimLocationKnown` $\in$ add$(a_j^{t_0})$. At the same time, Nurse 2 (N2) is exploring other areas. Once the victim's location is known, N2 can execute the action $a_i^{t_0+k}$ = `Navigate(Nurse Current Location, Victim Location)`, which has `VictimLocationKnown` $\in$ pre$(a_i^{t_0+k})$ as a precondition. Since only nurses are capable of treating victims, this coordination allows N2 to reach and assist V1.

- Giver Action: $a_j^{t_0}$ = `Interact(Firefighter, Victim)`
- Effect: `VictimLocationKnown` $\in$ add$(a_j^{t_0})$
- Receiver Action: $a_i^{t_0+k}$ = `Navigate(Nurse Current Location, Victim Location)`

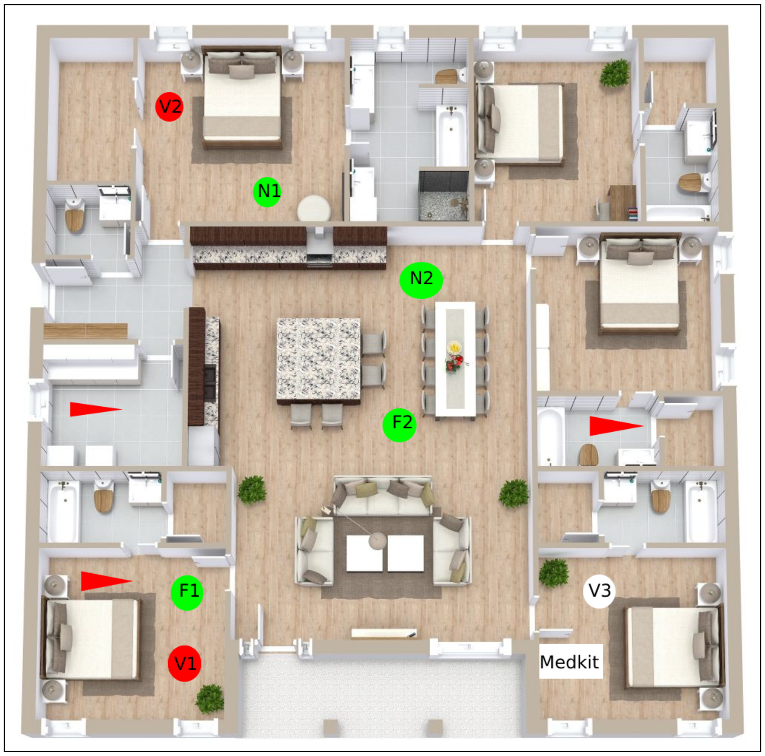


Figure 4: Illustration of an instance of the Search and Rescue Domain

- **Precondition:** `VictimLocationKnown` $\in$ pre$(a_i^{t_0+k})$
- Object: `Victim`

Example 2: Firefighter passes medical kit $\rightarrow$ Nurse treats victim : This scenario illustrates constructive sequential interdependence through the transfer of an object required for task completion. Nurse 1 (N1) needs a medical kit to treat Victim 2 (V2) but does not currently have one in their inventory and is located farther away from the kit. Firefighter 2 (F2), who is closer to the medical kit, performs the action $a_j^{t_0}$ = `Interact(Firefighter, MedicalKit, Nurse)`, resulting in the effect `Has(Nurse, MedicalKit)` $\in$ add$(a_j^{t_0})$. This enables N1 to subsequently perform the action $a_i^{t_0+k}$ = `Interact(Nurse, MedicalKit, Victim)`, which has `Has(Nurse, MedicalKit)` $\in$ pre$(a_i^{t_0+k})$ as a precondition. Since only nurses are capable of treating victims, F2's assistance is critical in enabling N1 help V2.

- Giver Action: $a_j^{t_0}$ = `Interact(Firefighter, MedicalKit, Nurse)`
- Effect: `Has(Nurse, MedicalKit)` $\in$ add$(a_j^{t_0})$
- Receiver Action: $a_i^{t_0+k}$ = `Interact(Nurse, MedicalKit, Victim)`
- Precondition: `Has(Nurse, MedicalKit)` $\in$ pre$(a_i^{t_0+k})$
- Object: `MedicalKit`

Example 3: Firefighter extinguishes fire $\rightarrow$ Nurse treats victim faster : This example highlights constructive interdependence where one agent modifies the environment to improve the effectiveness of another agent's action. In this scenario, Victim 3 (V3) is located in a room affected by fire, which hinders medical intervention. Nurse 2 (N2) is en route to treat the victim, but treatment is significantly faster and more effective if the fire has already been extinguished. Firefighter 1 (F1), who is in proximity to the fire, performs the action $a_j^{t_0}$ = `Interact(Firefighter, Extinguisher, Fire)`, resulting in the effect `FireExtinguished` $\in$ add($a_j^{t_0}$). This condition satisfies the precondition `FireExtinguished` $\in$ pre($a_i^{t_0+k}$) of the nurse's treatment action $a_i^{t_0+k}$ = `Interact(Nurse, MedicalKit, Victim)`, thereby enabling faster and more efficient treatment. This form of interdependence ensures that F1's timely intervention directly enhances N2's ability to save the victim.

- Giver Action: $a_j^{t_0}$ = `Interact(Firefighter, Extinguisher, Fire)`
- Effect: `FireExtinguished` $\in$ add($a_j^{t_0}$)
- Receiver Action: $a_i^{t_0+k}$ = `Interact(Nurse, MedicalKit, Victim)`
- Precondition: `FireExtinguished` $\in$ pre($a_i^{t_0+k}$) (for fast treatment)
- Object: `Fire`

These sequential interdependencies are goal-reaching and non-looping.

9.4 Looping vs. Non-Looping Sequential Interdependence

In our framework, sequential interdependencies $(a_i^{t_0+k}, a_j^{t_0})$ are defined as *goal-reaching* if the interaction contributes to final reward acquisition (e.g., successful victim treatment), and *non-looping* if the associated object obj^{int} is not returned to the original agent in the same state. That is, the influence trajectory $\tau_{\text{obj}}^{t_0}$ must be strictly progressing toward a terminal effect and not cyclic with respect to the state of obj^{int}. To illustrate a *looping interdependence*, consider the case where a police agent transfers a medical kit to a nurse at time t_0, and at time $t_0 + k$, the nurse returns the **same** kit to the police. If the state of the medical kit—denoted s(`MedicalKit`)—remains unchanged (e.g., `unused`, `intact`, `full-capacity`), and the kit does not contribute to any further task completion, then this constitutes a *looping* and *non-goal-reaching* interdependence. It is redundant and does not affect the task reward. Here, we consider a more nuanced scenario: At time t_0, the *nurse agent* transfers a `MedicalKit` to the *police agent* temporarily to free up their inventory (e.g., under an assumption that the nurse can initiate victim treatment bare-handed). At a later time $t_0 + k$, the police agent returns the same `MedicalKit` to the nurse, who then uses it to **complete** the victim treatment. In this case:

- The interdependence is *goal-reaching* since the treatment concludes successfully with enhanced reward.
- However, it is *looping*, as the object returns to its original holder in the same nominal state.

To resolve the case of *useful transfers*, we modify the state of the `MedicalKit` by augmenting it with a *usage-linked attribute*, such as:

$$s(\texttt{MedicalKit}) = \begin{cases} \texttt{unused} \\ \texttt{used-for-treatment} \\ \texttt{passed-temporarily} \end{cases}$$

By tagging the medical kit's state based on the context in which it was transferred (e.g., part of a treatment pipeline for a victim), we can distinguish *constructive looping interdependence* from *useful transfers*. This allows us to retain goal-relevant looping interdependencies while discarding non-contributing loops.

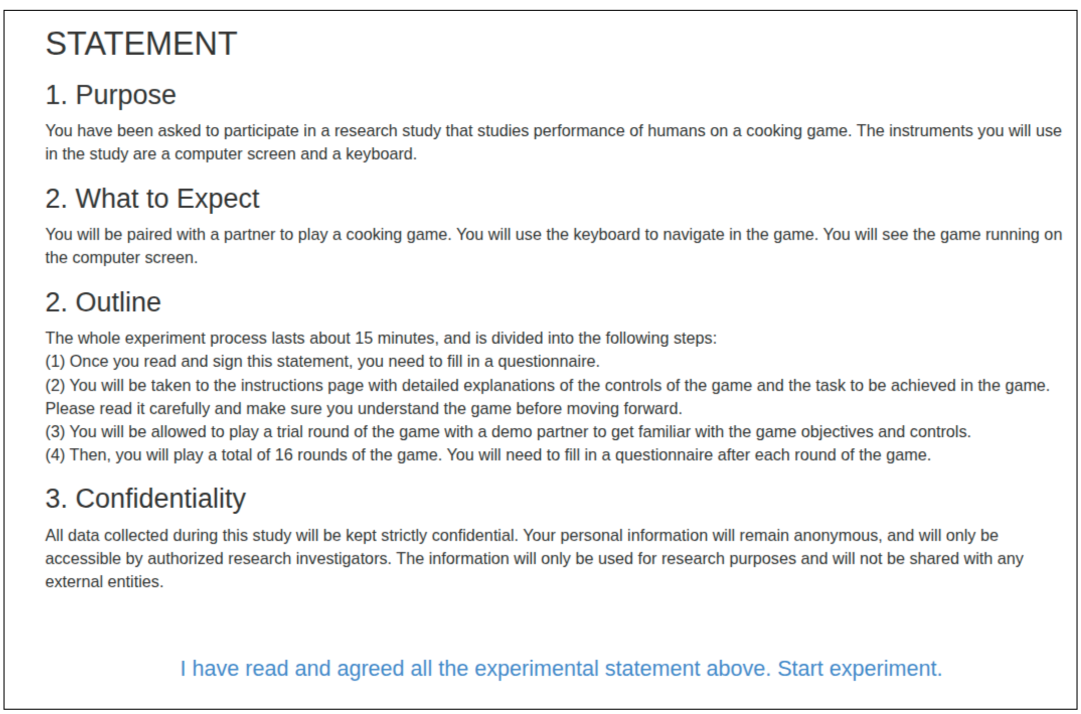
10 User Study Design:

We conducted a user study to evaluate the performance of state-of-the-art zero-shot coordination (ZSC) agents in a cooperative cooking game. The user study was built from Li et al. (2024; 2023); Sarkar et al. (2022). The purpose of this study was to understand how well these AI agents coordinate with human partners in real-time gameplay. Below we describe the study design, participants, game environments, agent details, and data collection process.

10.1 Consent and Experimental Statement

Each participant began the study by reviewing and agreeing to a consent statement. The statement explained the goals of the study, what participants would be asked to do, and how their data would be handled.

- **Purpose:** Participants were asked to take part in a study evaluating human performance when playing a cooperative cooking game with an AI partner.
- **Instruments:** The game was played using a computer screen and a keyboard.
- **Procedure:**
  1. After agreeing to the statement, participants filled out a demographic questionnaire.
  2. They read detailed instructions on how the game worked, including controls, rules, and objectives.
  3. They played a trial round with a scripted agent to become familiar with gameplay.
  4. They then played 16 rounds, each with a different pretrained AI partner.
  5. After each round, they filled out a short post-game questionnaire.
- **Confidentiality:** All data collected was kept confidential and anonymized. No personally identifiable information was stored or shared.



STATEMENT

1. Purpose

You have been asked to participate in a research study that studies performance of humans on a cooking game. The instruments you will use in the study are a computer screen and a keyboard.

2. What to Expect

You will be paired with a partner to play a cooking game. You will use the keyboard to navigate in the game. You will see the game running on the computer screen.

2. Outline

The whole experiment process lasts about 15 minutes, and is divided into the following steps:
(1) Once you read and sign this statement, you need to fill in a questionnaire.
(2) You will be taken to the instructions page with detailed explanations of the controls of the game and the task to be achieved in the game. Please read it carefully and make sure you understand the game before moving forward.
(3) You will be allowed to play a trial round of the game with a demo partner to get familiar with the game objectives and controls.
(4) Then, you will play a total of 16 rounds of the game. You will need to fill in a questionnaire after each round of the game.

3. Confidentiality

All data collected during this study will be kept strictly confidential. Your personal information will remain anonymous, and will only be accessible by authorized research investigators. The information will only be used for research purposes and will not be shared with any external entities.

I have read and agreed all the experimental statement above. Start experiment.

Figure 5: Consent statement shown to participants before starting the study.

CITI PROGRAM

Completion Date 14-May-2023
Expiration Date 14-May-2027
Record ID 55564726

This is to certify that:

[redacted]

Not valid for renewal of certification through CME.

Has completed the following CITI Program course:

Human Research
(Curriculum Group)
IRB – Social & Behavioral Research (Group 2)
(Course Learner Group)
1 - Basic Course
(Stage)

Under requirements set by:

[redacted]

CITI
Collaborative Institutional Training Initiative
101 NE 3rd Avenue, Suite 320
Fort Lauderdale, FL 33301 US
www.citiprogram.org

Verify at www.citiprogram.org/verify/?wf8a0bae8-061a-45c2-b678-5f82f7ef3705-55564726

10.2 IRB Certification for this User Study

10.3 Game Instructions and Layouts

Participants were introduced to the game rules and controls through an instruction page. The game involves two players (one human, one AI), cooking and serving onion soup. Each round involved coordination to serve a single soup within 60 seconds.

We used two layout types in our evaluation:

- Counter Circuit: Players can perform independent tasks with minimal interference.
- Forced Coordination: A layout that restricts movement and requires players to coordinate, making collaboration essential.

10.4 AI Partners and Evaluation

SOTA Methods: We evaluated four zero-shot coordination agents — FCP Strouse et al. (2022), MEP Zhao et al. (2022b), HSP Yu et al. (2023), and COLE Li et al. (2024). All these methods were trained using a two-stage framework:

- **Stage 1:** A diverse partner population is created through self-play.
- **Stage 2:** The ego agent is trained by playing against sampled partners from the population and optimizing task rewards using reinforcement learning.

Each approach differs in how partner diversity is encouraged:

- **FCP:** Direct self-play-based partner generation.
- **MEP:** Adds a maximum entropy term to encourage behavioral diversity in partners.

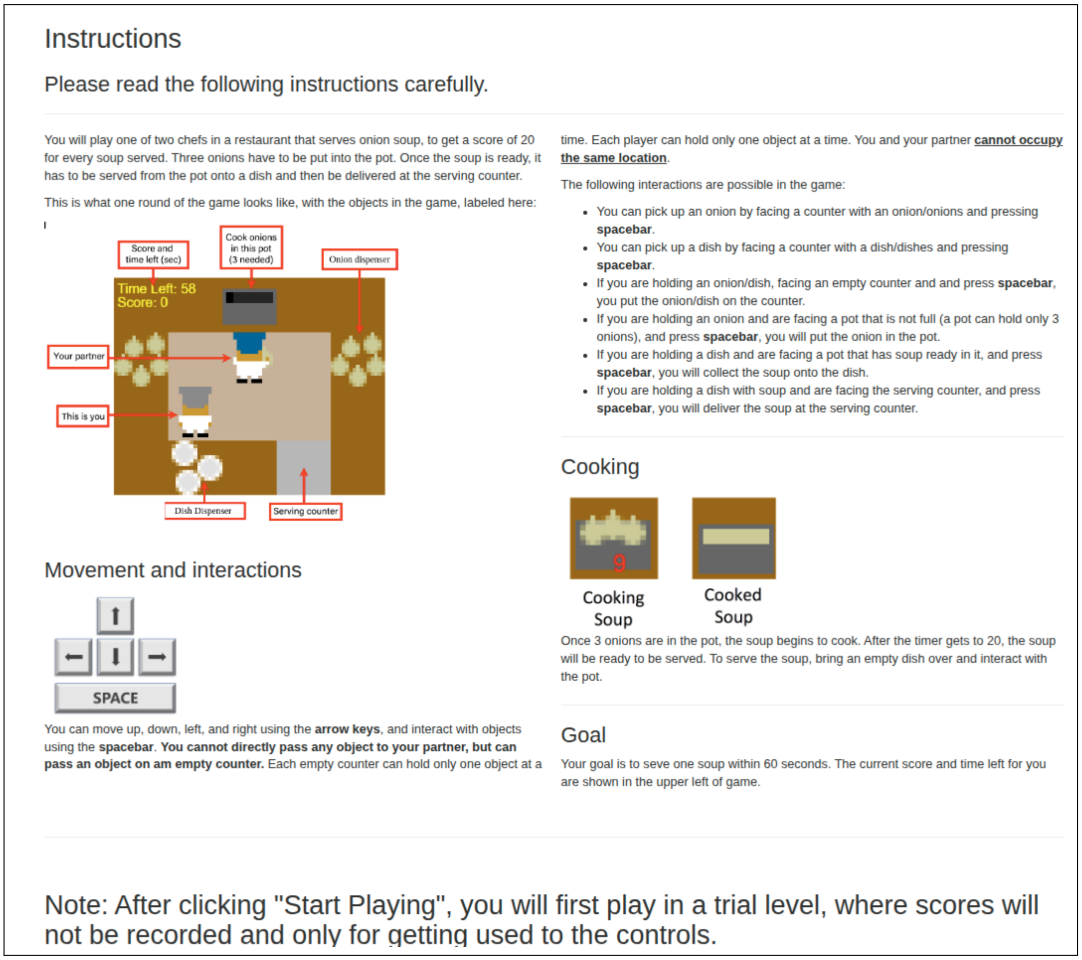


Figure 6: Instructions page shown to participants.

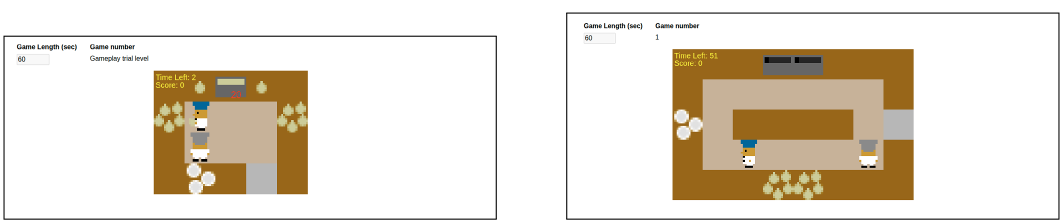


Figure 7: Left: Trial round gameplay with scripted partner. Right: Real round gameplay with SOTA AI partner.

- **HSP:** Constructs agents that model human preferences using event-based rewards.
- **COLE:** Treats the game as a graphical-form cooperative game, with rewards based on cooperative incompatibility distributions.

The ego agent in each case is evaluated based on episodic task reward while paired with a human partner.

10.5 Participants

We recruited 36 participants aged between 18 to 31 from our university, with a median age of 22.5 and an average age of 20.75. Out of these, 24 participants identified as male and 12 as female. A majority (63.9%) reported no prior familiarity with the Overcooked game. We conducted a pilot with 5 participants, and then used feedback from it to refine the final study with 31 participants.

10.6 Post-Round Questionnaire

After each round, participants filled out a questionnaire assessing collaboration, perceived responsiveness, and mutual intent. Each question was answered using a 5-point Likert scale (from "Strongly Disagree" to "Strongly Agree").

- **Team Performance:**
  - Q1. My partner and I worked together to deliver the soups.
  - Q2. My partner contributed to the successful delivery of the soups.
- **Were you working with your partner?**
  - Q3. I attempted to work with my partner to deliver the soups.
  - Q4. My partner responded to my attempts to work with them.
- **Was your partner working with you?**
  - Q5. My partner attempted to work with me.
  - Q6. I responded to their attempts to work with them.

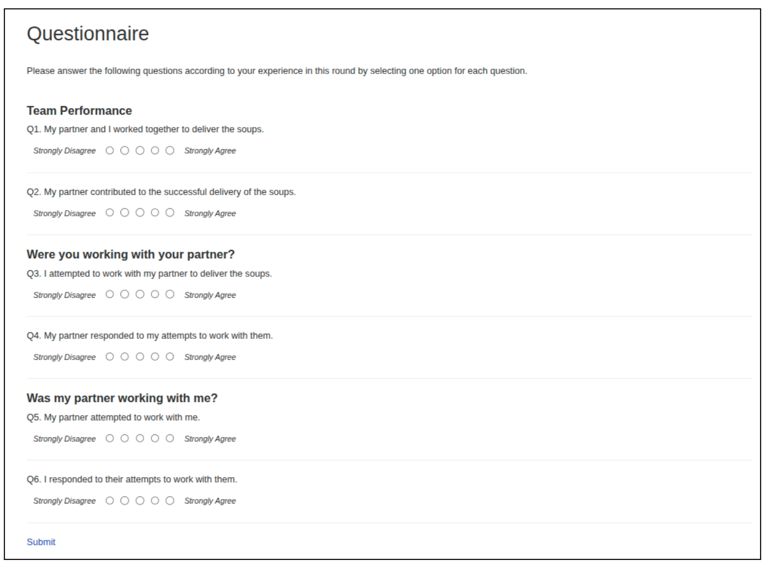


Figure 8: Post-round questionnaire interface shown to participants.

10.7 Questionnaire Design

Evaluating teamwork purely through task-based performance (e.g., reward or completion time) can miss nuanced aspects of coordination, intent, and mutual understanding — particularly in zero-shot collaboration scenarios. In this study, we developed an objective metric - **interdependence**, to measure the quality of team performance and cooperation in human-AI teams. The questionnaire was therefore designed to provide **additional subjective insights** into how humans perceived their AI partner's behavior. The central goal of our user study is to evaluate whether AI agents are capable of effective cooperation with human partners in zero-shot settings. Specifically, we want to assess two critical aspects of cooperative behavior:

- **Responsiveness:** Does the AI agent recognize and respond to the human's attempts to collaborate?
- **Proactiveness:** Does the AI agent initiate behaviors that attempt to induce or enable cooperation from the human partner?

This enabled us to answer a core research question: *Do the trajectories that score well under the interdependence metric also align with human perceptions of effective teamwork?* This alignment — or misalignment — between subjective and objective measures of teaming can reveal important gaps in AI-agent design, particularly in cooperative settings where behavior must be interpretable, responsive, and intuitive to humans. Each questionnaire was administered after a single round of gameplay and asked participants to reflect on their experience with that round's AI partner. The questions were grouped into three conceptual categories:

- **Team Performance (Q1, Q2):** These items measure whether the participant felt the round involved joint effort and contribution from both teammates toward the goal of delivering soup.
- **Agent Responsiveness to Participant Coordination (Q3, Q4):** Evaluates how effectively the agent responds when the participant initiates coordination.
- **Agent-Initiated Coordination and Participant Response (Q5, Q6):** Assesses how often the agent initiates coordination and how well these attempts are received by the participant.turn.

Each question was answered on a 5-point Likert scale (from *Strongly Disagree* to *Strongly Agree*). This design was inspired by constructs in human-robot interaction and team cognition research, such as perceived shared agency, responsiveness, and mutual intention. The repeated structure across 16 gameplay rounds allowed us to collect a rich set of human-AI interaction trajectories paired with subjective labels.

10.8 Statistical Tests

We tested the following null hypotheses related to participants' subjective perceptions of cooperation with their AI partners:

- *Counter Circuit Layout:*
  - $H_0^{1.1}$: The mean response to the statement *"My partner responded to my attempts to work with them"* equals the neutral midpoint (i.e., mean $= 3$).
  - $H_0^{1.2}$: The mean response to the statement *"My partner attempted to work with me"* equals the neutral midpoint (i.e., mean $= 3$).
- *Comparison Between Layouts (Counter Circuit vs. Forced Coordination):*
  - $H_0^{2.1}$: There is no difference in mean responses to *"My partner responded to my attempts to work with them"* between the two layouts (i.e., mean difference $= 0$).
  - $H_0^{2.2}$: There is no difference in mean responses to *"My partner attempted to work with me"* between the two layouts (i.e., mean difference $= 0$).

Formally, these hypotheses were tested using one-sample t-tests against the neutral midpoint for individual layouts, and paired t-tests for within-subject comparisons across layouts. For Q4, regarding partner responsiveness, responses in the Counter Circuit layout yielded a mean rating of 3.33. The one-sample t-test rejected the null hypothesis ($t(168) = 3.04$, $p = 0.0027$), indicating that participants perceived their partners as responding to their cooperation attempts at a level significantly above neutral. When comparing the two layouts within participants, a paired t-test showed a statistically significant difference ($t(23) = -2.24$, $p = 0.0352$), with higher perceived responsiveness reported in the Forced Coordination layout. Similarly, for Q5, which captures perceptions of partner initiative to cooperate, the Counter Circuit responses averaged 3.31. The one-sample t-test again rejected the null hypothesis of neutrality ($t(168) = 2.80$, $p = 0.0057$), suggesting that participants generally agreed their partners attempted to work with them.

Taken together, these subjective ratings suggest that, on average, participants felt their AI partners both responded to and attempted to cooperate with them. However, it is important to contextualize these findings within the broader experimental setting of zero-shot cooperation, where agents were paired with human participants exhibiting diverse behaviors — some actively seeking cooperation, while others preferred to act independently. This is reflected in objective measures, such as the average value of $\%H^{\text{trig}}_{\text{tot-sub}}$, which reveal that not all human participants wanted to engage in the cooperative strategy. These results underscore a key limitation of relying solely on subjective reports to evaluate cooperation: although participants generally perceive that their partners respond and attempt to work with them, this perception does not necessarily indicate that human-agent teams actually follow cooperative strategies. Objective behavioral analyses demonstrate that these teams did not do acooperative strategies, highlighting the importance of complementing subjective feedback with rigorous quantitative metrics when assessing human-agent collaboration.