Open Ad Hoc Teamwork with Cooperative Game Theory

Jianhong Wang [1] **Yang Li** [1] **Yuan Zhang** [2] **Wei Pan** [1] **Samuel Kaski** [1 3]

Abstract

Ad hoc teamwork poses a challenging problem, requiring the design of an agent to collaborate with teammates without prior coordination or joint training. Open ad hoc teamwork (OAHT) further complicates this challenge by considering environments with a changing number of teammates, referred to as open teams. One promising solution in practice to this problem is leveraging the generalizability of graph neural networks to handle an unrestricted number of agents with various agent-types, named graph-based policy learning (GPL). However, its joint Q-value representation over a coordination graph lacks convincing explanations. In this paper, we establish a new theory to understand the representation of the joint Q-value for OAHT and its learning paradigm, through the lens of cooperative game theory. Building on our theory, we propose a novel algorithm named CIAO, based on GPL's framework, with additional provable implementation tricks that can facilitate learning. The demos of experimental results are available on https://sites.google.com/view/ciao2024, and the code of experiments is published on https://github.com/hsvgbkhgbv/CIAO.

1. Introduction

Multi-agent reinforcement learning (MARL) has achieved partial success on multiple tasks including playing strategy games (Rashid et al., 2020), power system operation (Wang et al., 2021), and dynamic algorithm configuration (Xue et al., 2022). These tasks fit to the training paradigm of MARL, which requires all agents to be controllable and to be coordinated during training. However, with this paradigm it is difficult to tackle many real-world tasks where not all agents are controllable and even prior coordination may not be possible. For example, in search and rescue, a robot must collaborate with other robots it has not seen before (e.g., manufactured by various companies without a common coordination protocol) or humans to rescue survivors (Barrett & Stone, 2015). Similar situations occur in AI that helps trading markets (Albrecht & Ramamoorthy, 2013), as well as in the human-machine and machine-machine collaboration emerging from the prevailing embodied AI settings (Smith & Gasser, 2005; Duan et al., 2022) and large language models (Brown et al., 2020; Zhao et al., 2023).

To tackle the ad hoc teamwork problem, we explore a scenario where one agent, referred to as the **learner**, operates under our control and seeks to collaborate without prior coordination with **teammates** which have unknown types and policies (Stone et al., 2010). When dealing with teams of dynamic sizes, commonly termed **open teams**, the research problem addressed in this paper is often referred to as **open ad hoc teamwork** (OAHT) (Mirsky et al., 2022). One promising solution for OAHT in practice is graph-based policy learning (GPL) (Rahman et al., 2021). GPL presents a three-fold framework, encompassing a type inference model, a joint action value model and an agent model. Although GPL reaps the success of performance due to the generalizability of graph neural networks to handle an unrestricted number of agents with various agent-types, its weakness is that the representation of the joint Q-value over a coordination graph lacks convincing explanations. This restricts its applicability to real-world problems requiring trustworthy algorithms (Bhat & Alqahtani, 2021; Wang et al., 2021).

We propose to describe OAHT using a game model from cooperative game theory, namely the **coalitional affinity game** (CAG) (Brânzei & Larson, 2009). Specifically, we extend the CAG by incorporating Bayesian games (Harsanyi, 1967) to depict uncertain agent-types and stochastic games (Shapley, 1953) to represent the long-horizon goal. The resulting game is termed the **open stochastic Bayesian coalitional affinity game** (OSB-CAG). In this game, the learner aims to influence other teammates (via its actions) to collaborate in achieving a shared goal. To formalize this, we extend the standard cooperative game theory notion of strict core to a novel solution concept which we call **dynamic variational strict core** (DVSC). The DVSC transforms collaboration in a temporary team into the task of forming a stable temporary

[1]Center for AI Fundamentals, University of Manchester, UK [2]Neurorobotics Lab, University of Freiburg, Germany [3]Aalto University, Finland. Correspondence to: Jianhong Wang <jianhong.wang@manchester.ac.uk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

team, where no agent has incentives to leave. We model the OAHT process under the learner's influence as a **dynamic affinity graph** (equivalent to a coordination graph), generalizing the classical static CAG. Based on the dynamic affinity graph, we further conceptualize an agent's preference for a temporary team to measure whether they prefer to stay in the team under the learner's influence. GPL's joint action value model is proven to be the sum of any temporary agents' preferences over a long horizon.

The main contributions of this paper can be summarized as follows: (1) We conceptualize OAHT as a dynamic coalitional affinity game, OSB-CAG. In this model, the learner seeks to influence teammates through its actions, without prior coordination, to establish a stable temporary team. (2) The theoretical model of OSB-CAG gives an understanding of GPL's joint action value model. It ensures collaboration within any temporary team under open team settings. (3) Building on the OSB-CAG theory, we derive a constraint for representing the joint action value to facilitate learning, and an additional regularization term depending on the graph structure to rationalize solving DVSC as an RL problem. The novel algorithm, named CIAO (**C**ooperative game theory **I**nspired **A**d hoc teamwork in **O**pen teams), is implemented based on GPL and incorporates the above novel and provable tricks. (4) We discuss and understand the learning paradigm employed in GPL that aims to learn the joint Q-value for open team settings. (5) We conduct experiments, primarily comparing two instances of CIAO (CIAO-S and CIAO-C) based on GPL framework in two environments: Level-based Foraging (LBF) and Wolfpack under open team settings (Rahman et al., 2021). Finally, we conduct a comprehensive review and discussion of related works on both theoretical and algorithmic aspects of AHT and explore its relationship to MARL in Appendix A.

2. Background

Let $\Delta(\Omega)$ indicate the set of probability distributions over a random variable on a sample space Ω and let $\mathbb{P}(\mathcal{X})$ denote the power set of an arbitrary set $\mathcal{X}$. To simplify the notation, let i exclusively denote the learner and $-i$ denote the set of all temporary teammates at any timestep. $P(\mathcal{X})$ indicates the generic probability distribution over a random variable $\mathcal{X}$ and $|\mathcal{X}|$ indicates the cardinality of an arbitrary set $\mathcal{X}$.

2.1. Coalitional Affinity Game

As a subclass of non-transferable utility games, *hedonic game* (Chalkiadakis et al., 2022) is defined as a tuple $\langle \mathcal{N}, \geq \rangle$, where $\mathcal{N}$ is a set of all agents; and $\geq = (\geq_1, ..., \geq_n)$ is a sequence of agents' preferences over the subsets of $\mathcal{N}$ called coalitions. $\mathcal{C} \geq_j \mathcal{C}'$ implies that coalition $\mathcal{C}$ is no less preferred by agent j than coalition $\mathcal{C}'$. For each agent $j \in \mathcal{N}$, $\geq_j$ describes a complete and transitive preference relation over a collection of all feasible coalitions $\mathcal{N}(j) = \{\mathcal{C} \subseteq \mathcal{N} \mid j \in \mathcal{C}\}$. The outcome of a hedonic game is a coalition structure $\mathcal{CS}$, i.e., a partition of $\mathcal{N}$ into disjoint coalitions. We denote by $\mathcal{CS}(j)$ the coalition including agent j. The ordinal preferences can be represented as the cardinal form with preference values (Sliwinski & Zick, 2017). More specifically, an agent j has a preference value function such that $v_j : \mathcal{N}(j) \to \mathbb{R}_{\geq 0}$. $v_j(\mathcal{C}) \geq v_j(\mathcal{C}')$ if $\mathcal{C} \geq_j \mathcal{C}'$, which implies that agent j weakly prefers $\mathcal{C}$ to $\mathcal{C}'$; $v_j(\mathcal{C}) > v_j(\mathcal{C}')$ if $\mathcal{C} >_j \mathcal{C}'$, which implies that agent j strictly prefers $\mathcal{C}$ to $\mathcal{C}'$.

To concisely represent the preference value, a hedonic game is equipped with an affinity graph $G = \langle \mathcal{N}, \mathcal{E} \rangle$, where each edge $(j,k) \in \mathcal{E}$ describes an affinity relation between agents j and k. For each edge (j,k), it defines an *affinity weight* $w(j,k) \in \mathbb{R}$ to indicate the value that agent j can receive from agent k, while if $(j,k) \notin \mathcal{E}$, $w(j,k) = 0$. For any coalition $\mathcal{C} \subseteq \mathcal{N}_j$, the preference value of agent j is specified as $v_j(\mathcal{C}) = \sum_{(j,k)\in\mathcal{E}, k\in\mathcal{C}} w(j,k)$ if $\mathcal{C} \neq \{j\}$, otherwise, $v_j(\{j\}) = b_j \in \mathbb{R}_{\geq 0}$.[1] *An affinity graph is symmetric if* $w(j,k) = w(k,j)$, *for all* $(j,k), (k,j) \in \mathcal{E}$. The hedonic game with an affinity graph is named as *coalitional affinity game* (CAG) (Brânzei & Larson, 2009). Strict core stability is a principal solution concept of CAG (see Definition 1).

Definition 1. *We say that a blocking coalition $\mathcal{C}$ weakly blocks a coalition structure $\mathcal{CS}$ if every agent $j \in \mathcal{C}$ weakly prefers $\mathcal{C}$ to $\mathcal{CS}(j)$ and there exists at least one agent $k \in \mathcal{C}$ who strictly prefers $\mathcal{C}$ to $\mathcal{CS}(j)$. A coalition structure admitting no weakly blocking coalition $\mathcal{C} \subseteq \mathcal{N}$ is called strict core stable.*

2.2. Graph-Based Policy Learning

We now briefly review GPL's empirical framework (Rahman et al., 2021) to solve OAHT (see Appendix C.1 for more details). GPL consists of the following modules: the type inference model, the joint action value model and the agent model. To align with our motivation, we transform the framework to be adaptable to any coordination graph structure, as opposed to being restricted to only the complete graph as in GPL.

Type Inference Model. This is modelled as a LSTM (Hochreiter & Schmidhuber, 1997) to infer agent-types of a team at timestep t given the teammates' agent-types and the state at timestep $t-1$. The agent-type is modelled as a fixed-length hidden-state vector of LSTM, referred to as agent-type embedding. To address the issue of variable team size, the embedding of the agents who leave a team would be removed at each timestep, while the type embedding of the newly added agents would be set to a zero vector.

[1] In the original CAG setting (Sliwinski & Zick, 2017), $v_j(\{j\})$ is conventionally set to zero. Herein, we extend it to non-negative values for generality (see Appendix E).

Joint Action Value Model. The joint Q-value $\hat{Q}^{\pi^i}(s_t, a_t)$ is approximated as the sum of the individual utility $\hat{Q}_j^{\pi^i}(a_t^j, |s_t)$ and pairwise utility $\hat{Q}_{jk}^{\pi^i}(a_t^j, a_t^k|s_t)$:

$$\hat{Q}^{\pi^i}(s_t, a_t) = \sum_{(j,k)\in\mathcal{E}_t} \hat{Q}_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) + \sum_{j\in\mathcal{N}_t} \hat{Q}_j^{\pi^i}(a_t^j|s_t), \tag{1}$$

where the superscript π^i implies that the above terms can only be optimized by the learner's policy π^i.

Agent Model. To address the open team setting, GNN is applied to process the joint agent-type embedding θ_t produced from the type inference model, where each agent is represented as a node and the coordination graph is consistent with that for the joint action value model. The resulting node representation $\bar{n}_t$ is applied as input to infer the estimated teammates' joint policy, denoted as $\hat{\pi}^{-i}(a_t^{-i}|s_t, \theta_t^{-i})$.

Learner's Decision Making. The learner's approximate action value function $\hat{Q}^{\pi^i}(s_t, a_t^i)$ is defined as follows:

$$\hat{Q}^{\pi^i}(s_t, a_t^i) = \mathbb{E}_{a_t^{-i}\sim\pi_t^{-i}}\left[\hat{Q}^{\pi^i}(s_t, a_t^i, a_t^{-i})\right], \tag{2}$$

where s_t is a state at timestep t, a_t^{-i} is a joint action of teammates $-i$ at timestep t and a_t^i is the learner i's action at timestep t. The learner's decision making is conducted by selecting the action that maximizes $\hat{Q}^{\pi^i}(s_t, a_t^i)$.

3. A New Game Model to Formalize OAHT

In this section, we generalize the coalitional affinity game framework to formalize OAHT, by integrating a graph to represent relationships among agents. It is essential to emphasize that, for the sake of brevity, our focus of this work is exclusively on fully observable scenarios.

3.1. Problem Formulation

In an environment, the learner i interacts with other uncontrollable temporary teammates $-i$ to achieve a shared goal. To model this process, we introduce Open Stochastic Bayesian Coalitional Affinity Game (OSB-CAG), defined as a tuple $\langle \mathcal{N}, \mathcal{S}, (\mathcal{A}_j)_{j\in\mathcal{N}}, \Theta, (R_j)_{j\in\mathcal{N}}, P_T, P_I, P_A, \mathcal{E}, \gamma\rangle$. Here, $\mathcal{N}$ represents the set of all possible agents, $\mathcal{S}$ is the set of states, $\mathcal{A}_j$ is the action set for agent j, and Θ denotes the set of all possible agent-types. Let the joint action set under a variable agent set $\mathcal{N}_t \subseteq \mathcal{N}$ be defined as $\mathcal{A}_{\mathcal{N}_t} = \times_{j\in\mathcal{N}_t}\mathcal{A}_j$. Therefore, the joint action space under a variable number of agents is defined as $\mathcal{A}_{\mathcal{N}} = \bigcup_{\mathcal{N}_t\in\mathbb{P}(\mathcal{N})}\{a|a \in \mathcal{A}_{\mathcal{N}_t}\}$, while the joint agent-type space under a variable number of agents is defined as $\Theta_{\mathcal{N}} = \bigcup_{\mathcal{N}_t\in\mathbb{P}(\mathcal{N})}\{\theta|\theta \in \Theta^{|\mathcal{N}_t|}\}$. A dynamic affinity graph, denoted as $G_t = \langle \mathcal{N}_t, \mathcal{E}_t\rangle$, is introduced to describe the relationships among agents. Here, $\mathcal{E}_t = \{(j,k) \mid j,k \in \mathcal{N}_t\} \subseteq \mathcal{E}$, and $\mathcal{E}$ is a set of possible edges represented by pairs (j,k). This graph is referred to as the coordination graph in GPL.

Transition Function. We now introduce three primitive probability distributions denoted as $P_T : \mathbb{P}(\mathcal{N}) \times \mathcal{S} \times \mathcal{A}_{\mathcal{N}} \to \Delta(\mathbb{P}(\mathcal{N}) \times \mathcal{S})$, $P_I : \mathbb{P}(\mathcal{N}) \times \mathcal{S} \to [0,1]$, and $P_A : \mathcal{N} \times \mathcal{S} \to \Delta(\Theta)$. These probability functions characterize the dynamics of the environment in the following procedure: (1) At the initial timestep 0, $P_I(\mathcal{N}_0, s_0)$ generates an initial set of agents $\mathcal{N}_0$ and an initial state s_0. (2) $P_A(\theta_t^j|\{j\}, s_t)$ represents a type assignment function that randomly assigns agent-types to the generated agent set. (3) $P_T(\mathcal{N}_t, s_t|\mathcal{N}_{t-1}, s_{t-1}, a_{t-1})$ generates the agent set $\mathcal{N}_t$ and state s_t for the next time step t. (4) Stage 2 and 3 above are repeated. To succinctly represent the aforementioned process, we derive a composite transition function $T(\mathcal{N}_t, s_t, \theta_t|s_{t-1}, a_{t-1}, \theta_{t-1})$ (see Proposition 1) in place of stage 2 and 3 from timesteps $t \geq 1$. This function can be factorized, clarifying the GPL's framework, as follows:

$$\begin{aligned} &T(\mathcal{N}_t, s_t, \theta_t|s_{t-1}, a_{t-1}, \theta_{t-1}) \\ &\quad = P_E(\theta_t|\mathcal{N}_t, s_t)P_O(\mathcal{N}_t, s_t|s_{t-1}, a_{t-1}, \theta_{t-1}). \end{aligned} \tag{3}$$

Herein, $P_O(\mathcal{N}_t, s_t|s_{t-1}, a_{t-1}, \theta_{t-1})$ is a probability distribution composed of P_T, P_I and P_A (see the sketch of proof of Proposition 1) that generates a variable agent set $\mathcal{N}_t$ and a state s_t, *observable* by the learner. In contrast, a joint agent-type θ_t generated from $P_E(\theta_t|\mathcal{N}_t, s_t) = \prod_{j=1}^{|\mathcal{N}_t|} P_A(\theta_t^j|\{j\}, s_t)$ is *unobservable* by the learner. However, it plays a crucial role in the agent model for the learner's decision making in the empirical framework of GPL, motivating the estimation of this term in practice, as conducted by the type inference model (see Section 2.2). To distinguish between and clarify the observation generated from P_O and the agent-types generated from P_E during the decision process, both functions will be concurrently utilized to describe the composite transition function T in the subsequent sections. To simplify the notation, we would use P_O in place of P_I for $t = 0$ in the following sections.

Assumption 1. *The following conditional independencies are assumed to hold in any distribution P over the set of variables in an OSB-CAG: (1) $(\theta_t \perp\!\!\!\perp \theta_{t-1}, s_{t-1}, a_{t-1} \mid \mathcal{N}_t, s_t)$; (2) $(\mathcal{N}_t, s_t \perp\!\!\!\perp \theta_{t-1} \mid \mathcal{N}_{t-1}, s_{t-1}, a_{t-1})$; (3) $(\mathcal{N}_t \perp\!\!\!\perp a_t|s_t, \theta_t)$; (4) $(\theta_t^j \perp\!\!\!\perp -j, \theta_t^{-j} \mid \{j\}, s_t)$.*

Proposition 1. *$T(\mathcal{N}_t, s_t, \theta_t|s_{t-1}, a_{t-1}, \theta_{t-1})$ for $t \geq 1$ can be expressed in terms of the following well-defined probability distributions: $P_I(\mathcal{N}_0, s_0)$, $P_T(\mathcal{N}_t, s_t|\mathcal{N}_{t-1}, s_{t-1}, a_{t-1})$ for $t \geq 1$, and $P_A(\theta_t^j|\{j\}, s_t)$ for $t \geq 0$.*

Proof. We show the sketch of proof here. The following derivation is obtained by Assumption 1. For validity of conditions in Assumption 1, please refer to Appendix D. About the complete version of proof, please refer to Appendix G.1.

$$\begin{aligned} &T(\mathcal{N}_t, s_t, \theta_t|s_{t-1}, a_{t-1}, \theta_{t-1}) = \\ &\qquad P_E(\theta_t|\mathcal{N}_t, s_t)P_O(\mathcal{N}_t, s_t|s_{t-1}, a_{t-1}, \theta_{t-1}), \end{aligned}$$

where $P_E(\theta_t|\mathcal{N}_t, s_t) = \prod_{j=1}^{|\mathcal{N}_t|} P_A(\theta_t^j|\{j\}, s_t)$ and

$$P_O(\mathcal{N}_t, s_t|s_{t-1}, a_{t-1}, \theta_{t-1}) = \sum_{\mathcal{N}_{t-1}} P_T(\mathcal{N}_t, s_t|\mathcal{N}_{t-1}, s_{t-1}, a_{t-1}) P(\mathcal{N}_{t-1}|s_{t-1}, \theta_{t-1}).$$

We have

$$P(\mathcal{N}_t|s_t, \theta_t) = \frac{\sum_{s_t} P_E(\theta_t|\mathcal{N}_t, s_t) P(\mathcal{N}_t, s_t)}{\sum_{\mathcal{N}_t} \sum_{s_t} P_E(\theta_t|\mathcal{N}_t, s_t) P(\mathcal{N}_t, s_t)}.$$

Also, we have $P(\mathcal{N}_0, s_0) = P_I(\mathcal{N}_0, s_0)$ and when $t \geq 1$,

$$P(\mathcal{N}_t, s_t) = \sum_{\mathcal{N}_{t-1}} \sum_{s_{t-1}} \sum_{a_{t-1}} \sum_{\theta_{t-1}} P(\mathcal{N}_t, s_t, \mathcal{N}_{t-1}, s_{t-1}, a_{t-1}, \theta_{t-1}),$$

where $P(\mathcal{N}_t, s_t, \mathcal{N}_{t-1}, s_{t-1}, a_{t-1}, \theta_{t-1})$ can be expressed in terms of $P(\mathcal{N}_{t-1}, s_{t-1})$ and the probability distributions we have defined. The sketch of proof is completed. □

Preference Reward. The function $R_j : \mathcal{A}_\mathcal{N} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$ extends an agent j's preference value, of the original stateless CAG, to the agent j's preference reward R_j which depends on the state and action. For example, $R_j(a_t|s_t)$ indicates agent j's preference reward for a temporary team $\mathcal{N}_t \subseteq \mathcal{N}$ with the corresponding joint action $a_t = \times_{j \in \mathcal{N}_t} a_t^j$, whereas $R_j(a_t^j|s_t)$ indicates agent j's preference reward for a coalition only including itself. To capture the relationship between agents j and k in terms of both the current state and the actions taken, the affinity weight is generalized accordingly as $w_{jk} : \mathcal{A}_j \times \mathcal{A}_k \times \mathcal{S} \to \mathbb{R}$. Following the specification of preference values through affinity weights, the preference reward of any agent j for a coalition $\mathcal{N}_t$ can be represented as $R_j(a_t|s_t) = \sum_{(j,k) \in \mathcal{E}_t, k \in \mathcal{N}_t} w_{jk}(a_t^j, a_t^k|s_t)$. This summation aggregates the affinity weights for all pairs of agents (j, k) in the coalition, where k is a member of $\mathcal{N}_t$. The learner's reward function $R(s_t, a_t)$ for any $\mathcal{N}_t$ is specified by $R_j(a_t|s_t)$, which will be introduced in Section 3.4.

3.2. Dynamic Variational Strict Core

We now extend the game theoretical concept of strict core from CAG to OSB-CAG as a criterion to evaluate the extent of collaboration among the agents in a temporary team (a coalition $\mathcal{N}_t$ at each timestep t), named as *dynamic variational strict core* (DVSC). Unlike the strict core defined in CAG that evaluates coalition formation based on the given preference values, DVSC evaluates whether the learner i's policy can influence temporary teammates' decisions (measured by preference rewards), so that they intend to collaborate (so called variational). This is analogous to forming a temporary team as a desired coalition. Next we derive a result on strict core stability to motivate a result on DVSC. The following two statements are equivalent when the affinity graph is *symmetric*: Team maximizes social welfare, and team reaches strict core stability (see Lemma 1 in Appendix F). This inspires using the objective of maximizing social welfare as a surrogate criterion to evaluate strict core stability, and this criterion can be further generalized to dynamic scenarios to derive the DVSC (see Definition 2).

Definition 2. *If a dynamic affinity graph is symmetric, then maximizing the long-horizon social welfare is equivalent to reaching strict core stability under the variable teammates of uncertain agent-types generated by P_E and the uncertain states generated by P_O.*

Following the inspiration shown in Definition 2, DVSC can be equivalently expressed in the form shown in Eq. (4). The detailed derivation of DVSC is left in Appendix F.

$$\texttt{DVSC} := \Big\{ \pi^{i,*} \,\Big|\, \mathbb{E}_{\pi^{i,*}}\Big[\sum_{t=0}^{\infty} \gamma^t \sum_{j \in \mathcal{N}_t} R_j(a_t|s_t)\Big] \geq \mathbb{E}_{\pi^i}\Big[\sum_{t=0}^{\infty} \gamma^t \sum_{j \in \mathcal{N}_t} R_j(a_t|s_t)\Big], \forall s_0 \in \mathcal{S}, \forall \pi^i \Big\}, \tag{4}$$

where $a_t^i \sim \pi^i$ and $a_t^{-i} \sim \pi_t^{-i}$; $\mathbb{E}_{\pi^i}[\cdot]$ denotes the expectation that also implicitly depends on $\theta_t \sim P_E$ and $\mathcal{N}_t, s_t \sim P_O$, and $a_t^{-i} \sim \pi_t^{-i}$; and $\pi^{i,*}$ indicates the solution to DVSC.

3.3. Is Stability of any Temporary Team a Reasonable Metric for Describing Open Ad Hoc Collaboration?

Recall that all agents in AHT have a shared goal, which implies that they intrinsically aim to collaborate on solving a shared task (Mirsky et al., 2022), but their preferences for collaborating with each other are not necessarily compatible. This compatibility can be interpreted as stability of a temporary team, determined by the preferences of ad hoc agents for collaborating with each other. If those ad hoc agents are incompatible with each other, the temporary team becomes unstable but still with hope of collaborating as a team to solve the shared task. Thus, the learner's aim is to tweak the compatibility of a temporary team through its actions, to influence the temporary teammates' preferences, equivalent to maintaining the stability of the temporary team, across timesteps. Furthermore, the definition of DVSC in Eq. (4) is invariant to team size, aligned to open team settings.

3.4. Solving DVSC by Reinforcement Learning

We proceed to define the learner's reward function, initially left blank in Section 3.1 and convert DVSC from Eq. (4) into an RL problem. Since the learner's objective is to execute actions that influence any temporary teammates to collaboratively solve a shared task, we naturally interpret the learner's reward function as $R(s_t, a_t) = \sum_{j \in \mathcal{N}_t} R_j(a_t|s_t)$. The reward function represents the social welfare of preference rewards for a temporary team $\mathcal{N}_t$, serving as a metric to measure agents' preferences for collaborating on a shared

task.[2] Substituting $R(s_t, a_t)$ into Eq. (4), we derive an RL problem equivalent to solving DVSC:

$$\max_{\pi_i} \mathbb{E}_{\mathcal{N}_t, s_t \sim P_O, \theta_t \sim P_E, a_t^{-i} \sim \pi_t^{-i}, a_t^i \sim \pi^i} \Big[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \Big]. \quad (5)$$

In the following section, we will explore how the optimization problem in Eq. (5) can be solved by a novel algorithm.

4. A Novel Algorithm Building on OSB-CAG

In this section, we derive a novel graph-based RL algorithm to solve OAHT based on the OSB-CAG, with DVSC as a solution concept. We first derive the joint Q-value's representation to narrow down its hypothesis space including the solution of DVSC. The representation aligns with and gives an interpretation to the GPL's heuristic joint action value model. Note that we also acquire a condition to further confine the joint Q-value's hypothesis space thanks to our theory (see Section 4.1). With the estimated type inference model and agent model, the optimal learner's policy obtained by GPL's optimization problem approximately reaches DSVC (see Section 4.2). Finally, we derive a novel practical algorithm, named CIAO (see Section 4.3).

4.1. Representation of Joint Q-Value

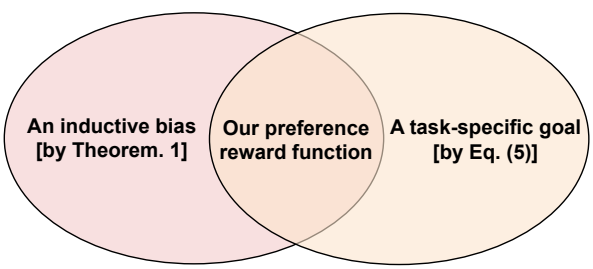


Figure 1: Illustration of the relationship between the conditions for our preference reward function, ensuring the existence of DVSC under its confined hypothesis space, and its alignment to a task-specific reward $R(s_t, a_t)$ in Eq. (5).

Given the joint actions generated under the influence by the optimal learner's policy $\pi^{i,*}$, we have a sufficient condition, as an inductive bias, for *any preference reward function* to narrow down its hypothesis space meeting DVSC in Theorem 1. Solving the RL problem outlined in Eq. (5) based on this condition to specify $\pi^{i,*}$, the preference reward function is aligned to a task-specific reward $R(s_t, a_t)$. The relationship between the above conditions to generate our preference reward function is shown in Fig. 1.

Theorem 1. *In an OSB-CAG, for any dynamic affinity graph $G_t = \langle \mathcal{N}_t, \mathcal{E}_t \rangle$ at any timestep t, if there exists a joint action $a_t \in \mathcal{A}_{\mathcal{N}_t}$, for any agent $j \in \mathcal{N}_t$, satisfying $R_j(a_t|s_t) \geq R_j(a_t^j|s_t)$ for any $s_t \in \mathcal{S}$, then DVSC always exists.*

To meet the condition that $R_j(a_t|s_t) \geq R_j(a_t^j|s_t)$ as shown in Theorem 1, we derive a representation of $w_{jk}(a_t^j, a_t^k|s_t)$ in Proposition 2. Recall that an agent j's preference reward function for a temporary team $\mathcal{N}_t$ at timestep t is defined as $R_j(a_t|s_t) = \sum_{(j,k)\in\mathcal{E}_t} w_{jk}(a_t^j, a_t^k|s_t)$ (see Section 3.1).

Proposition 2. *In a dynamic affinity graph $G_t = \langle \mathcal{N}_t, \mathcal{E}_t \rangle$, for any state $s_t \in \mathcal{S}$ and any joint action $a_t \in \mathcal{A}_{\mathcal{N}_t}$, if for all $(j,k) \in \mathcal{E}_t$, $w_{jk}(a_t^j, a_t^k|s_t) = \alpha_{jk}(a_t^j, a_t^k|s_t) + \beta_{jk}(a_t^j|s_t)$ with the conditions that $\alpha_{jk}(a_t^j, a_t^k|s_t) \geq 0$ and $R_j(a_t^j|s_t) = \sum_{(j,k)\in\mathcal{E}_t} \beta_{jk}(a_t^j|s_t)$, then $R_j(a_t|s_t) \geq R_j(a_t^j|s_t)$ for any agent $j \in \mathcal{N}_t$.*

Proof. This result can be directly obtained by the definition that $R_j(a_t|s_t) = \sum_{(j,k)\in\mathcal{E}_t, k\in\mathcal{N}_t} w_{jk}(a_t^j, a_t^k|s_t)$. □

Plugging in the expression of $w_{jk}(a_t^j, a_t^k|s_t)$, we can obtain the representation of an arbitrary agent j's preference Q-value under the learner's optimal policy $\pi^{i,*}$, $Q_j^{\pi^{i,*}}(a_t|s_t) = \sum_{(j,k)\in\mathcal{E}_t} Q_{jk}^{\pi^{i,*}}(a_t^j, a_t^k|s_t) + Q_j^{\pi^{i,*}}(a_t^j|s_t)$, and the joint Q-value under the learner's optimal policy $\pi^{i,*}$, $Q^{\pi^{i,*}}(s_t, a_t) = \sum_{j\in\mathcal{N}_t} Q_j^{\pi^{i,*}}(a_t|s_t)$, outlined in Theorem 2.

Assumption 2. *Suppose that $\alpha_{jk}(a_t^j, a_t^k|s_t) = 0$ for $t \geq T$, where T is the timestep when agent j or k leaves the environment, and $R_j(a_t^j|s_t) = 0$ for $t \geq T'$, where T' is the timestep when agent j leaves the environment.*

Theorem 2. *Under Assumption 2, if $w_{jk}(s_\tau, a_\tau^j, a_\tau^k) = \alpha_{jk}(s_\tau, a_\tau^j, a_\tau^k) + \beta_{jk}(s_\tau, a_\tau^j)$, then the joint Q-value of the learner's policy π^i can be expressed as follows:*

$$\begin{aligned} Q^{\pi^i}(s_t, a_t) &= \sum_{(j,k)\in\mathcal{E}_t} Q_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) + \sum_{j\in\mathcal{N}_t} Q_j^{\pi^i}(a_t^j|s_t) \\ &= \sum_{j\in\mathcal{N}_t} \Big\{ \sum_{(j,k)\in\mathcal{E}_t} Q_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) + \sum_{j\in\mathcal{N}_t} Q_j^{\pi^i}(a_t^j|s_t) \Big\} \\ &:= \sum_{j\in\mathcal{N}_t} Q_j^{\pi^i}(a_t|s_t), \end{aligned}$$

where $Q_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) = \mathbb{E}_{\pi^i}[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} \alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau)]$ and $Q_j^{\pi^i}(a_t^j|s_t) = \mathbb{E}_{\pi^i}[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} R_j(a_\tau^j|s_\tau)]$.

Remark 1. *The result of Theorem 2 verifies that the optimal joint Q-value representation derived from our theory is consistent with the GPL's joint action value model, as shown in Eq. (1), but additionally with $\hat{Q}_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) \geq 0$, following our theory, which is requisite for satisfying $\alpha_{jk}(a_t^j, a_t^k|s_t) \geq 0$, as shown in Proposition 2.*

Recall that the condition for solving DSVC as a RL problem is *the symmetry of a dynamic affinity graph* (see Definition 2). To meet this condition, we outline in Proposition 3 the

[2] In practical scenarios, $R(s_t, a_t)$ only needs to implicitly encode the shared goal that multiple agents are required to achieve.

constraints that must be fulfilled for the case of a dynamic affinity graph being a star graph (see Remark 2 for its validity in OAHT). Similarly, we provide the relevant constraints articulated in Proposition 4 for situations where the dynamic affinity graph takes the form of a complete graph (as applied to GPL). The implementation of the constraints for these two cases are shown in Remark 3.

Definition 3. *In this paper, we introduce a novel dynamic affinity graph structured as a star graph, with the learner serving as the internal node and temporary teammates as the leaf nodes.*

Remark 2. *We introduce a novel architecture for the dynamic affinity graph in the context of OAHT, assuming teammates lack prior coordination (Mirsky et al., 2022). Given an additional assumption that teammates cannot adapt their policies or types in response to other agents,*[3] *it is reasonable to presume the absence of relationships among any temporary teammates. Besides, this is also in line with the assumption in AHT that the learner's temporary teammates might not be familiar with one another before the interaction (Stone et al., 2010; Mirsky et al., 2022). In particular, this implies that no edges between any two teammates are necessary to form a dynamic affinity graph. However, the learner's goal is to establish collaboration with a variable number of temporary teammates at each timestep, necessitating the existence of edges between the learner and each teammate. To meet all these requirements, we design the learner's dynamic affinity graph as a star graph, as detailed in Definition 3. Consequently, the preference reward of any teammate j for a temporary team $\mathcal{N}_t$ is determined as $R_j(s_t, a_t) = w_{ji}(s_t, a_t^j, a_t^i)$, while the learner i's preference reward for the temporary team $\mathcal{N}_t$ is expressed as $R_i(s_t, a_t) = \sum_{j \in -i} w_{ij}(s_t, a_t^i, a_t^j)$.*

Proposition 3. *For the learner i and any teammate j or k, the constraints $R_i(a_t^i|s_t) = \sum_{j \in -i} R_j(a_t^j|s_t)$ and $\alpha_{jk}(a_t^j, a_t^k|s_t) = \alpha_{kj}(a_t^k, a_t^j|s_t)$, for any $a_t \in \mathcal{A}_{\mathcal{N}_t}$ and $s_t \in \mathcal{S}$, are necessary for a star dynamic affinity graph to be symmetric.*

Proposition 4. *For any two agents j or k, the constraints $R_j(a_t^j|s_t) = R_k(a_t^k|s_t)$ and $\alpha_{jk}(a_t^j, a_t^k|s_t) = \alpha_{kj}(a_t^k, a_t^j|s_t)$, for any $a_t \in \mathcal{A}_{\mathcal{N}_t}$ and $s_t \in \mathcal{S}$, are necessary for the complete dynamic affinity graph to be symmetric.*

Remark 3. *The following implementation is necessary to satisfy the symmetry of a dynamic affinity graph: (1) meeting $Q_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) = Q_{kj}^{\pi^i}(a_t^k, a_t^j|s_t) \geq 0$ in constructing preference Q-values; (2) If the dynamic affinity graph is a star graph with the learner as the internal node, $Q_i^{\pi^i}(a_t^i|s_t) = \sum_{j \in -i} Q_j^{\pi^i}(a_t^j|s_t)$ is implemented as a regularizer. If the dynamic affinity graph is a complete graph, $Q_i^{\pi^i}(a_t^i|s_t) = Q_j^{\pi^i}(a_t^j|s_t)$ is implemented as a regularizer.*

[3] For simplicity in presenting our theory in this paper, we tentatively disregard scenarios where temporary teammates can adapt to other agents (e.g. establishing an affinity model).

4.2. Bellman Optimality Equation for OSB-CAG

We now define the Bellman optimality equation for OSB-CAG to evaluate the learner i's optimal policy $\pi^{i,*}$ as a solution of the DVSC following Theorem 3, such that

$$Q^{\pi^{i,*}}(s_t, a_t) = R(s_t, a_t) + \gamma \mathbb{E}_{\mathcal{N}_{t+1}, s_{t+1} \sim P_O}\Big[\max_{a^i} \mathbb{E}_{\theta_{t+1} \sim P_E,\ a_{t+1}^{-i} \sim \pi_{t+1}^{-i}}\big[Q^{\pi^{i,*}}(s_{t+1}, a_{t+1}^{-i}, a^i)\big]\Big]. \tag{6}$$

The regularity condition of Eq. (6) is that $\mathcal{N}_{t+1} \subseteq \mathcal{N}_t$, since it is pathological to consider an agent $j \in \mathcal{N}_{t+1}$ but, $\notin \mathcal{N}_t$ at timestep t when expanding $Q^{\pi^{i,*}}(s_t, a_t)$ across timesteps, which is clarified in an illustrative example in Fig. 2.

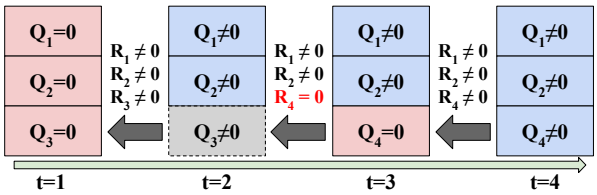


Figure 2: Illustration of the expansion of Bellman optimality equation for OSB-CAG. The thin green arrow indicates the time axis, while the thick black arrow indicates the expansion direction of Bellman optimality equation. In the theory of OSB-CAG, we have $Q^{\pi^{i,*}}(s_t, a_t) = \sum_{j \in \mathcal{N}_t} Q_j^{\pi^{i,*}}(a_t|s_t)$, where $Q_j^{\pi^{i,*}}$ is denoted by $\mathbf{Q}_j$ in the figure. R_j indicates the preference reward of agent j to the team, measuring the agent's preference to stay in the team to solve the shared task. At each timestep, the preference Q-value $\mathbf{Q}_j$ of an agent j that joins the game is filled in a red box, that remains in the game is filled in a blue box, and that leaves the game filled in a grey box with the dashed outline. At timestep 1, agent 1, 2 and 3 just join the game, so with all preference Q-values as zeros. At timestep 2, agent 3 leaves the game, and the expansion works as usual, since agent 3 has influence to the team. At timestep 3, agent 4 joins the game, but it has not any influence to the team. For this reason, it is unnecessary to consider the expansion for agent 4's preference Q-value *only at timestep* 3, since it would be trivially zero. To satisfy the more generic representation of the joint Q-value with respect to preference Q-values (other than the linear decomposition described in our theory), we rule out the transition samples of $\mathcal{N}_t \subset \mathcal{N}_{t+1}$. At timestep 4, the expansion considers all existing agents as usual.

Theorem 3. *Under Assumption 2 and an arbitrary learner's deterministic stationary policy π^i, the Bellman equation for the OSB-CAG with DVSC as a solution concept is expressed as follows: $Q^{\pi^i}(s_t, a_t) = R(s_t, a_t) + \gamma \mathbb{E}_{\mathcal{N}_{t+1}, s_{t+1} \sim P_O}\Big[\mathbb{E}_{\theta_{t+1} \sim P_E,\ a_{t+1} \sim \pi_{t+1}}\big[Q^{\pi^i}(s_{t+1}, a_{t+1})\big]\Big]$.*

To solve Eq. (6), we further propose an operator with the same regularity condition, such that $\Gamma : Q \mapsto \Gamma Q$, specified as follows:

$$\Gamma Q^{\pi^i}\left(s_{t+1}, a^{-i}_{t+1}, a^i\right) := R(s_t, a_t) + \gamma \mathbb{E}_{\mathcal{N}_{t+1}, s_{t+1}\sim P_O}\Big[\max_{a^i} \mathbb{E}_{\theta_{t+1}\sim P_E,\ a^{-i}_{t+1}\sim \pi^{-i}_{t+1}}\big[Q^{\pi^i}(s_{t+1}, a^{-i}_{t+1}, a^i)\big]\Big]. \tag{7}$$

Eq. (7) is a standard form of Bellman operator. Therefore, recursively running Eq. (7) converges to the Bellman optimality equation in Eq. (6), following the well-known value iteration algorithm (Sutton & Barto, 2018, Ch. 4).

Remark 4. *In implementation, the effect of $\mathcal{N}_t \subset \mathcal{N}_{t+1}$ can be omitted, due to its low proportions during the process. Therefore, solving the GPL optimization problem of fitted Q-learning (Ernst et al., 2005) that omits the effect of $\mathcal{N}_t \subset \mathcal{N}_{t+1}$ is a reasonable approximation of Bellman operator in Eq. (7), which reduces the computational cost of filtering out the transition samples of $\mathcal{N}_t \subset \mathcal{N}_{t+1}$ in practice. The GPL optimization problem is shown as follows:*

$$\min_{\beta} L(\beta) = \mathbb{E}\Big[\frac{1}{2}\Big(R(s_t, a_t) + \gamma \max_{a^i} \mathbb{E}_{\substack{\theta_{t+1}\sim P_E,\\ a^{-i}_{t+1}\sim \pi^{-i}_{t+1}}}\big[\hat{Q}^{\pi^i}(s_{t+1}, a^{-i}_{t+1}, a^i; \beta^-)\big] - \hat{Q}^{\pi^i}(s_t, a_t; \beta)\Big)^2\Big], \tag{8}$$

where $\hat{Q}^{\pi^i}(\cdot\,;\,\beta^-)$ is the approximate target optimal joint Q-value parameterised by β^- and $\hat{Q}^{\pi^i}(\cdot\,;\,\beta)$ is the approximate optimal joint Q-value parameterised by β.

4.3. Practical Implementation

Based on our theory, we introduce a novel algorithm, CIAO, representing the algorithm for **C**ooperative game theory **I**nspired **A**d hoc teamwork in **O**pen teams. We implement CIAO in dynamic affinity graphs as a star graph (refer to Remark 2 for more insights into this topology) and a complete graph, denoted as *CIAO-S* and *CIAO-C*, respectively, where "S" signifies **S**tar graph and "C" signifies **C**omplete graph. In addition to the joint Q-value representation model (derived from Theorem 2) and the training losses for estimating the unknown type inference model and the unknown agent model (as detailed in Section 2.2), we introduce novel Q losses tailored for variant dynamic affinity graphs based on our theory. These losses incorporate regularization terms with multipliers $\lambda > 0$.

CIAO-S. If the dynamic affinity graph is a star graph, the training loss with the regularizer is as follows:

$$L_s(\beta) = L(\beta) + \lambda \mathbb{E}_{s_t, a_t}\Big[\frac{1}{2}\Big(\sum_{j\in -i} \hat{Q}^{\pi^i}_j(a^j_t|s_t) - \hat{Q}^{\pi^i}_i(a^i_t|s_t;\beta)\Big)^2\Big].$$

CIAO-C. If the dynamic affinity graph is a complete graph, the training loss with the regularizer is as follows:

$$L_c(\beta) = L(\beta) + \lambda \mathbb{E}_{s_t, a_t}\Big[\sum_{j\in -i} \frac{1}{2}\big(\hat{Q}^{\pi^i}_i(a^i_t|s_t) - \hat{Q}^{\pi^i}_j(a^j_t|s_t;\beta)\big)^2\Big].$$

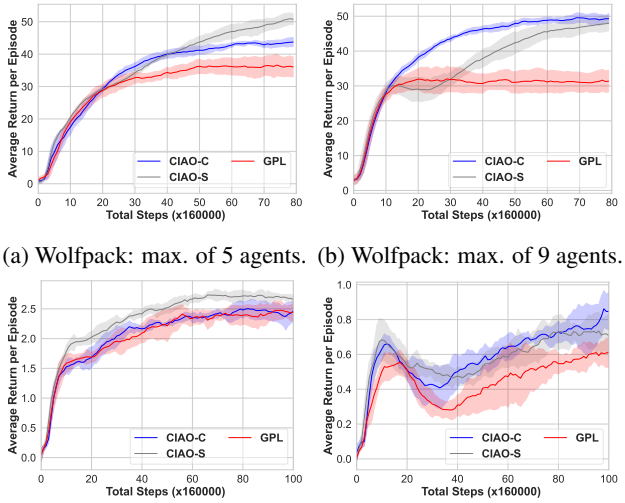
Note that it is also requisite to enforce that $\hat{Q}^{\pi^i}_{jk}(a^j_t, a^k_t|s_t) = \hat{Q}^{\pi^i}_{kj}(a^k_t, a^j_t|s_t) \geq 0$ by Remark 3. Following our theoretical model, the learner's reward $R(s_t, a_t)$ ought to be non-negative, while the designated reward of an environment could be negative. However, this can be adjusted by adding the maximum difference between these two rewards among states and joint actions denoted by $\Delta R(s_t, a_t)$ without changing the original goal. In practice, Eq. (8) is solved by DQN (Mnih et al., 2013). The learner's actions are decided by Eq. (2), employing the estimated teammates' agent models $\hat{\pi}^{-i}$ (see Section 2.2) to marginalize a^{-i}_t of $\hat{Q}^{\pi^i}(s_t, a_t; \beta)$, as implemented in GPL. The further implementation details are left to Appendix C.

5. Experiments

We assess the effectiveness of the proposed algorithms CIAO-S and CIAO-C in two established environments, LBF and Wolfpack, featuring open team settings (Rahman et al., 2021). In these settings, teammates are randomly selected to enter the environment and remain for a certain number of time steps. During experiments, the learner is trained in an environment with a maximum of 3 agents at each timestep. Subsequently, testing is conducted in environments with a maximum of 5 and 9 agents at each timestep, showcasing the model's ability to handle both unseen compositions and varied team sizes. All experiments are conducted with five random seeds, and the results are presented as the mean performance with a 95% confidence interval. Our experimental design aims to answer the following questions: (1) Does the joint Q-value representation outlined in our theory effectively facilitate collaboration between the learner and temporary teammates? (2) Is it necessary to generalize the preference reward function from zero, as in CAG, to a non-negative range in our theory (see Appendix E)? (3) Is the claim in Remark 4 valid in practice? (4) Is CIAO able to improve the generalization of agent-type sets?

Baselines and Ablation Variants. The state-of-the-art baseline we use in this experiment is GPL-Q (shortened as GPL) (Rahman et al., 2021). The ablation variants of the proposed CIAO are as follows: **CIAO-X-FI**, **CIAO-X-ZI** and **CIAO-X-NI** are variants that remove enforcement of individual utility, enforce individual utility as zero and enforce individual utility as negative values, respectively. **CIAO-X-NP** is a variant that enforces negative pairwise utility. "X" above indicates either "S" or "C". Further details on experimental settings can be found in Appendix H.

5.1. Main Results



(a) Wolfpack: max. of 5 agents. (b) Wolfpack: max. of 9 agents.

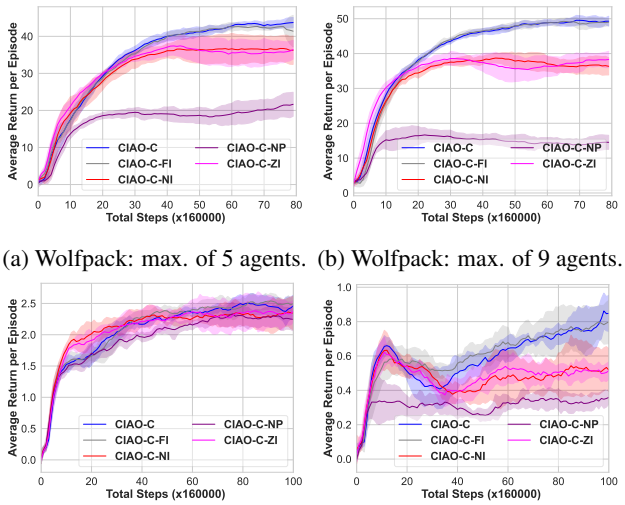
(c) LBF: max. of 5 agents. (d) LBF: max. of 9 agents.

Figure 3: Comparison between CIAO and GPL in Wolfpack and LBF with a maximum of 5 and 9 agents.

We initially address Question 1 through experiments conducted on the original versions of Wolfpack and LBF, as depicted in Fig. 3. It is evident that CIAO-C outperforms GPL in the majority of scenarios with varying maximum numbers of agents. This not only verifies the correctness and effectiveness of our theory, irrespective of dynamic affinity graph structures but also demonstrates its capability in facilitating collaboration between the learner and temporary teammates in the open ad hoc teamwork problem. Upon comparing CIAO-C and CIAO-S, it becomes apparent that the star graph may be more effective in scenarios with fewer agents, whereas the complete graph exhibits greater effectiveness in scenarios with more agents. This observation aligns with the intuition that the direct influence from the learner to each teammate may not suffice as the number of agents increases. Instead, indirect influence, where a teammate is influenced by the learner to subsequently influence another teammate, becomes crucial.

5.2. Ablation Study

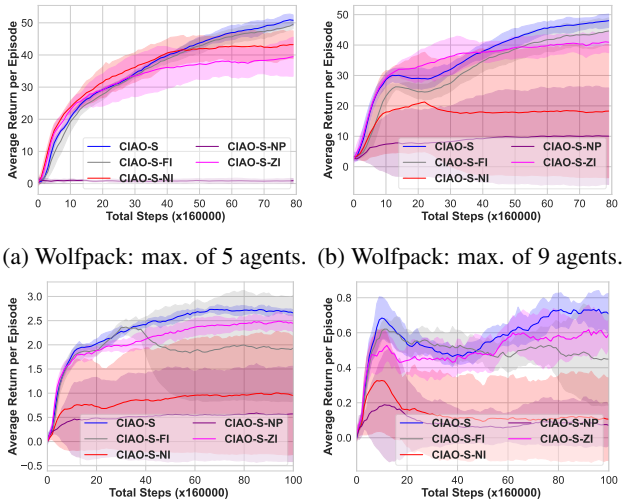
We present experimental results comparing CIAO-S and its ablations, as well as CIAO-C and its ablations. As illustrated in Figs. 4 and 5, both CIAO-C-NP and CIAO-S-NP exhibit notably inferior performance compared to CIAO-C or CIAO-S. This observation demonstrates the validity of DVSC and confirms the accuracy of the joint Q-value representation based on our theory. This outcome provides an additional perspective in addressing Question 1. Adhering to the tradition of CAG, convention mandates setting individual utility to zero. However, in our theory, we extend its range to include non-zero values, enhancing its adaptability across diverse scenarios. This adaptability is demonstrated in the comparison between CIAO-C or CIAO-S and CIAO-C-ZI or CIAO-S-ZI in Figs. 4 and 5. Although our theory does not inherently provide specific insights into the range of individual utility, we propose a hypothesis aligned with other definitions in CAG, asserting that individual utility is non-negative. This hypothesis ensures self-consistency in our generalization, as detailed in Definition 4 in Appendix E. The superior performances of CIAO-C or CIAO-S over their ablations affirm the acceptability of our hypothesis.



(a) Wolfpack: max. of 5 agents. (b) Wolfpack: max. of 9 agents.

(c) LBF: max. of 5 agents. (d) LBF: max. of 9 agents.

Figure 4: Comparison between CIAO-C and its ablations in Wolfpack and LBF with a maximum of 5 and 9 agents.

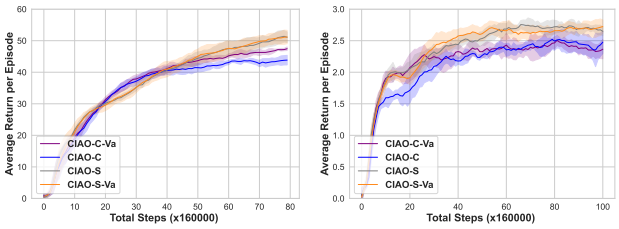


(a) Wolfpack: max. of 5 agents. (b) Wolfpack: max. of 9 agents.

(c) LBF: max. of 5 agents. (d) LBF: max. of 9 agents.

Figure 5: Comparison between CIAO-S and its ablations in Wolfpack and LBF with a maximum of 5 and 9 agents.

5.3. Validation for Remark 4



(a) Wolfpack: max. of 5 agents. (b) LBF: max. of 5 agents.

Figure 6: Comparison of training losses for CIAO between the implementations with omitting the effect of $\mathcal{N}_t \subset \mathcal{N}_{t+1}$ and those without (denoted as "-Va").

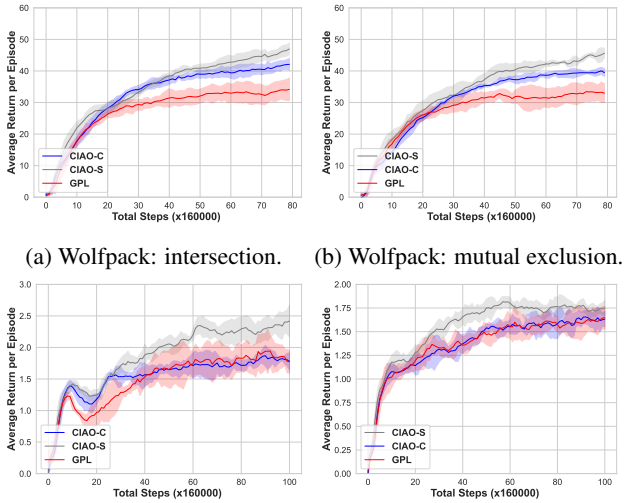
We now validate our claim in Remark 4 that minimizing the GPL training loss (omitting the effect of $\mathcal{N}_t \subset \mathcal{N}_{t+1}$) is an approximation of Eq. (7). Based on the GPL training loss, we implement its variant that filters out the transition samples of $\mathcal{N}_t \subset \mathcal{N}_{t+1}$, following the suggestion from Remark 4, referred to as CIAO-C-Va and CIAO-S-Va. As shown in Fig. 6, in both LBF and Wolfpack with the maximum of 5 agents, CIAO-C and CIAO-S trained with the GPL training loss achieve the approximate performances to those with the variant training loss considering the effect of $\mathcal{N}_t \subset \mathcal{N}_{t+1}$.

5.4. Generalization of Agent-Type Sets

We now evaluate the generalizability of CIAO to agent-type sets through two scenarios: (1) the agent-type set for training has intersection of one agent-type with that for testing; (2) the agent-type set for training is mutually exclusive to that for testing. As seen from Fig. 7, the dynamic affinity graph as the star graph is more generalizable than the complete graph. One hypothesis for this phenomenon is that although the complete graph may be able to capture broader relationships among agents, it could be unnecessary for open ad hoc teamwork (see Remark 2). The underlying principles behind this result deserve to be investigated in the future research. Although the generalizability of GNNs (which is also implemented in GPL) has featured prominently in generalization of agent-type sets (Rahman et al., 2021), the overall superior performance of CIAO to GPL still empirically shows CIAO's effectiveness to this problem.

6. Conclusion

Discussion. In this work we address the challenging problem of open ad hoc teamwork, aiming to design an agent capable of collaborating with teammates without prior coordination under dynamically changing team compositions. We propose a novel approach by incorporating cooperative game theory to develop a new theory. This theory effectively gives an interpretation to the joint Q-value representation



(a) Wolfpack: intersection. (b) Wolfpack: mutual exclusion.

(c) LBF: intersection. (d) LBF: mutual exclusion.

Figure 7: Comparison between CIAO and GPL on Wolfpack and LBF, with intersecting (denoted as "intersection") and mutually exclusive (denoted as "mutual exclusion") agent-type sets in training and testing, respectively. The maximum temporary team size is 5.

leveraged in the state-of-the-art algorithm, GPL. In addition, the learning paradigm employed in GPL's framework is understood based on our theory. Building upon the empirical foundation of GPL, we introduce a novel algorithm, CIAO, which includes an additional regularizer and a constraint for representation thanks to our theory. Consequently, CIAO can be seen as an extension of GPL, providing extra information through our theory to narrow down the joint Q-value's hypothesis space, to improve learning efficiency. Furthermore, the incorporation of dynamic affinity graphs into OSB-CAG opens up a new avenue of designing graphs describing agent relationships aligned to game objectives. Experimental results validate the effectiveness of our theory and demonstrate the superior performance of CIAO.

Limitation and Future Work. This work is the first in establishing both a theory and a practical algorithm rooted in cooperative game theory to address ad hoc teamwork. It opens up avenues of several promising future directions. Firstly, to enhance the scope and applicability of our theory, a logical next step involves exploring the adaptivity of teammates with time-varying agent-types, a factor currently omitted in our theory for simplicity. Another compelling direction is investigating the design of understandable joint Q-value representation for open ad hoc teamwork, other than linear decomposition with pairwise relationships and individual values justified in this work. This thread can push forward the potential deployment of ad hoc teamwork to safety-critical environments requiring trustworthy and cost-saving solutions, with fewer trial-and-error interactions.

Impact Statement

The outcomes of this paper could significantly enhance the progress of autonomous vehicles, smart grids, and various decision-making scenarios involving multiple independently controlled agents under uncertainties. However, it is crucial to acknowledge potential drawbacks. Like many machine learning algorithms, our work may encounter challenges related to human value alignment, when the targets in interaction are humans in the potential applications. Addressing this concern is part of our ongoing research, building upon findings from related fields that emphasize alignment issues.

Acknowledgement

This work is partially supported by UKRI Turing AI World-Leading Researcher Fellowship, EP/W002973/1. The computational resources are supported by CSC – IT Center for Science LTD., Finland. Yuan Zhang receives funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 953348 (ELO-X).

References

Agmon, N. and Stone, P. Leading ad hoc agents in joint action settings with multiple teammates. In *AAMAS*, pp. 341–348, 2012.

Agmon, N., Barrett, S., and Stone, P. Modeling uncertainty in leading ad hoc teams. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pp. 397–404, 2014.

Albrecht, S. V. and Ramamoorthy, S. A game-theoretic model and best-response learning method for ad hoc coordination in multiagent systems. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pp. 1155–1156, 2013.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Barrett, S. and Stone, P. Cooperating with unknown teammates in complex domains: A robot soccer case study of ad hoc teamwork. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Barrett, S., Rosenfeld, A., Kraus, S., and Stone, P. Making friends on the fly: Cooperating with new teammates. *Artificial Intelligence*, 242:132–171, 2017.

Bhat, J. R. and Alqahtani, S. A. 6g ecosystem: Current status and future perspective. *IEEE Access*, 9:43134–43167, 2021.

Brafman, R. I. and Tennenholtz, M. On partially controlled multi-agent systems. *Journal of Artificial Intelligence Research*, 4:477–507, 1996.

Brânzei, S. and Larson, K. Coalitional affinity games. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pp. 1319–1320, 2009.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chalkiadakis, G., Elkind, E., and Wooldridge, M. *Computational aspects of cooperative game theory*. Springer Nature, 2022.

Chen, S., Andrejczuk, E., Cao, Z., and Zhang, J. AATEAM: achieving the ad hoc teamwork by employing the attention mechanism. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 7095–7102. AAAI Press, 2020.

De Peuter, S. and Kaski, S. Zero-shot assistance in sequential decision problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11551–11559, 2023.

Du, Y., Han, L., Fang, M., Liu, J., Dai, T., and Tao, D. Liir: Learning individual intrinsic reward in multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Duan, J., Yu, S., Tan, H. L., Zhu, H., and Tan, C. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2):230–244, 2022.

Ernst, D., Geurts, P., and Wehenkel, L. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6, 2005.

Foerster, J., Assael, I. A., De Freitas, N., and Whiteson, S. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Gu, P., Zhao, M., Hao, J., and An, B. Online ad hoc teamwork under partial observability. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Harsanyi, J. C. Games with incomplete information played by "bayesian" players, i–iii part i. the basic model. *Management science*, 14(3):159–182, 1967.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Jiang, J. and Lu, Z. Learning attentional communication for multi-agent cooperation. *Advances in neural information processing systems*, 31, 2018.

Kalyanakrishnan, S., Liu, Y., and Stone, P. Half field offense in robocup soccer: A multiagent reinforcement learning case study. In *RoboCup 2006: Robot Soccer World Cup X 10*, pp. 72–85. Springer, 2007.

Kim, D., Moon, S., Hostallero, D., Kang, W. J., Lee, T., Son, K., and Yi, Y. Learning to schedule communication in multi-agent reinforcement learning. *arXiv preprint arXiv:1902.01554*, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Koller, D. and Friedman, N. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Mguni, D. H., Jafferjee, T., Wang, J., Nieves, N. P., Slumbers, O., Tong, F., Li, Y., Zhu, J., Yang, Y., and Wang, J. LIGS: learnable intrinsic-reward generation selection for multi-agent learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Mirsky, R., Carlucho, I., Rahman, A., Fosong, E., Macke, W., Sridharan, M., Stone, P., and Albrecht, S. V. A survey of ad hoc teamwork research. In *Multi-Agent Systems: 19th European Conference, EUMAS 2022, Düsseldorf, Germany, September 14–16, 2022, Proceedings*, pp. 275–293. Springer, 2022.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Rahman, M. A., Hopner, N., Christianos, F., and Albrecht, S. V. Towards open ad hoc teamwork using graph-based policy learning. In *International Conference on Machine Learning*, pp. 8776–8786. PMLR, 2021.

Rashid, T., Samvelyan, M., de Witt, C. S., Farquhar, G., Foerster, J. N., and Whiteson, S. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4292–4301. PMLR, 2018.

Rashid, T., Farquhar, G., Peng, B., and Whiteson, S. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Shapley, L. S. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.

Shneiderman, B. Human-centered artificial intelligence: Three fresh ideas. *AIS Transactions on Human-Computer Interaction*, 12(3):109–124, 2020.

Sliwinski, J. and Zick, Y. Learning hedonic games. In *IJCAI*, pp. 2730–2736, 2017.

Smith, L. and Gasser, M. The development of embodied cognition: Six lessons from babies. *Artificial life*, 11(1-2): 13–29, 2005.

Stone, P. and Kraus, S. To teach or not to teach? decision making under uncertainty in ad hoc teams. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pp. 117–124, 2010.

Stone, P., Kaminka, G. A., and Rosenschein, J. S. Leading a best-response teammate in an ad hoc team. In *International Workshop on Agent-Mediated Electronic Commerce*, pp. 132–146. Springer, 2009.

Stone, P., Kaminka, G. A., Kraus, S., and Rosenschein, J. S. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In Fox, M. and Poole, D. (eds.), *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press, 2010.

Sukhbaatar, S., Fergus, R., et al. Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 29, 2016.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V. F., Jaderberg, M., Lanctot, M., Sonnerat,

N., Leibo, J. Z., Tuyls, K., and Graepel, T. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pp. 2085–2087. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Tacchetti, A., Song, H. F., Mediano, P. A. M., Zambaldi, V. F., Kramár, J., Rabinowitz, N. C., Graepel, T., Botvinick, M. M., and Battaglia, P. W. Relational forward models for multi-agent learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Wang, J., Zhang, Y., Kim, T.-K., and Gu, Y. Shapley q-value: A local reward approach to solve global reward games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7285–7292, Apr 2020.

Wang, J., Xu, W., Gu, Y., Song, W., and Green, T. C. Multi-agent reinforcement learning for active voltage control on power distribution networks. *Advances in Neural Information Processing Systems*, 34:3271–3284, 2021.

Wang, J., Zhang, Y., Gu, Y., and Kim, T.-K. Shaq: Incorporating shapley value theory into multi-agent q-learning. *Advances in Neural Information Processing Systems*, 35: 5941–5954, 2022.

Wu, F., Zilberstein, S., and Chen, X. Online planning for ad hoc autonomous agent teams. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 439–445, 2011.

Xie, A., Losey, D., Tolsma, R., Finn, C., and Sadigh, D. Learning latent representations to influence multi-agent interaction. In *Conference on robot learning*, pp. 575–588. PMLR, 2021.

Xue, K., Xu, J., Yuan, L., Li, M., Qian, C., Zhang, Z., and Yu, Y. Multi-agent dynamic algorithm configuration. *Advances in Neural Information Processing Systems*, 35: 20147–20161, 2022.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

Zintgraf, L. M., Devlin, S., Ciosek, K., Whiteson, S., and Hofmann, K. Deep interactive bayesian reinforcement learning via meta-learning. In Dignum, F., Lomuscio, A., Endriss, U., and Nowé, A. (eds.), *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*, pp. 1712–1714. ACM, 2021. doi: 10.5555/3463952.3464210.

A. Related Works

Theoretical Models for Ad Hoc Teamwork. In our review of theoretical models for describing ad hoc teamwork (AHT), we begin by discussing foundational works. Brafman & Tennenholtz (1996) pioneered the study of ad hoc teamwork by investigating the repeated matrix game with a single teammate. Subsequent contributions extended this line of inquiry to scenarios involving multiple teammates, as exemplified by Agmon & Stone (2012), who expanded the analysis to incorporate multiple teammates. Agmon et al. (2014) further relaxed assumptions by allowing teammates' policies to be drawn from a known set. Stone et al. Stone & Kraus (2010) proposed collaborative multi-armed bandits, initially formalizing AHT but with notable assumptions, such as knowing teammates' policies and environments. Albrecht & Ramamoorthy (2013) introduced the stochastic Bayesian game (SBG) as the first complete theoretical model for addressing dynamic environments and unknown teammates in AHT. Building upon the SBG, Rahman et al. (2021) proposed the open stochastic Bayesian game (OSBG) to address open ad hoc teamwork (OAHT). Zintgraf et al. (2021) modelled AHT as interactive Bayesian reinforcement learning (IBRL) in Markov games, focusing on solving non-stationary teammates' policies within episodes. In contrast, Xie et al. (2021) introduced a hidden parameter Markov decision process (HiP-MDP) to address scenarios where teammates' policies vary across episodes but remain stationary within each episode. In this paper, we contribute to the theoretical landscape of AHT by extending the coalitional affinity game (CAG) from the perspective of cooperative game theory, under the assumptions similar to SBG and OSBG. In more details, we introduce a novel theoretical model, referred to as **O**pen **S**tochastic **B**ayesian **C**oalitional **A**ffinity **G**ame (OSB-CAG), shedding light on the interactive process between the learner and temporary teammates. This theoretical model can be seen as an extension of OSBG (see Appendix B), where the relationship between agents is conceptualized as a dynamic affinity graph in theory, moving beyond treating the graph solely as an implementation tool.[4] Our proposed solution concept, DVSC, provides a fresh perspective on how the learner can find optimal policies to attract temporary teammates for effective collaboration. Furthermore, we introduce a more specified transition function under our theoretical model in place of the one proposed by Rahman et al. (2021). The main benefit of our proposed transition function is that it enjoys a strong relationship to the underlying assumptions, and explicitly subsumes the concrete interactive process described by Rahman et al. (2021).

Algorithms for Ad Hoc Teamwork. We now review AHT from an algorithmic standpoint. The best response algorithm (Stone et al., 2009), initially proposed under the assumptions of a matrix game and well-known teammates' policies, laid the foundation for algorithmic solutions in this domain. Extending this work, REACT (Agmon et al., 2014) emerged as a solution effective for matrices where teammates' policies are drawn from a known set. Wu et al. (2011) introduced a novel approach using biased adaptive play to estimate teammates' actions based on their historical actions. They combined this with Monte Carlo tree search to plan the ad hoc agent's actions. HBA (Albrecht & Ramamoorthy, 2013) expanded the scope beyond matrix games, maintaining a probability distribution of predetermined agent-types and maximizing long-term payoffs through an extended Bellman operator. PLASTIC-Policy (Barrett et al., 2017) addressed more realistic scenarios, such as RoboCup (Kalyanakrishnan et al., 2007), by training teammates' policies through behavior cloning and the ad hoc agent's policy through FQI (Ernst et al., 2005). AATEAM (Chen et al., 2020) extended PLASTIC-Policy, incorporating an attention network (Bahdanau et al., 2014) to enhance the estimation of unseen agent-types. Rahman et al. (2021) integrated modern deep learning techniques, including GNNs and RL algorithms, into HBA to address open ad hoc teamwork (OAHT) and introduced GPL. ODITS (Gu et al., 2022) was proposed to handle teammates with rapidly changing behaviors under partial observability. In this paper, we introduce CIAO, a novel algorithm based on our proposed theory (OSB-CAG with DVSC as a solution concept). Specifically, CIAO extends the joint Q-value representation and training loss of GPL. Additionally, CIAO generalizes the implementation of training losses to various structures of the dynamic affinity graph, known as the coordination graph in GPL, with theoretical guarantees. This provides a design paradigm of training loss to facilitate the investigation of diverse dynamic affinity graph structures. This paradigm not only can cater for various scenarios of applications, but also can facilitate realizing the ideas inspired by other fields. Furthermore, we prove in theory and demonstrate in experiments that the existing GPL training loss is a viable approximation of the exact learning paradigm under our theory.

Relationship to Cooperative Multi-Agent Reinforcement Learning. Cooperative multi-agent reinforcement learning (MARL) primarily aims at training and controlling agents altogether to optimally achieve a shared goal. The key research topics are credit assignment (also known as value decomposition in some literature) (Foerster et al., 2018; Sunehag et al., 2018; Rashid et al., 2018), reward shaping (Du et al., 2019; Mguni et al., 2022), and communication (Foerster et al., 2016; Sukhbaatar et al., 2016; Jiang & Lu, 2018; Kim et al., 2019). In this paper, we shift the focus to AHT, where only one agent (referred to as learner) is controllable and trained to collaborate with an unknown set of uncontrollable agents to achieve

[4] If the dynamic affinity graph is with no edges, the OSB-CAG will degrade to a plain OSBG.

a shared goal. Although the teammates' behaviours in AHT can be influenced by the learner's action (under assumption that they are capable of reacting to the learner's action) (Mirsky et al., 2022), the joint policy may still be sub-optimal owing to either the reactivity of teammates or the effectiveness to attract teammates in implementation. On the other hand, a transferable utility game known as the convex game, belonging to cooperative game theory was introduced for employing Shapley value as a credit assignment scheme with theoretical guarantees and interpretation, to address credit assignment (Wang et al., 2020; 2022). In this paper, we introduce CAG, belonging to non-transferable utility games (a broader class including transferable utility games), for establishing a graph-based joint Q-value representation with theoretical guarantees and understandings to address OAHT.

B. Open Stochastic Bayesian Game

We now review the open stochastic Bayesian game (OSBG) that describes the open ad hoc teamwork for establishing GPL (Rahman et al., 2021). It is defined as a tuple such that $\langle \mathcal{N}, \mathcal{S}, (\mathcal{A}_j)_{j\in\mathcal{N}}, \Theta, R, T, \gamma \rangle$. $\mathcal{N}$ is a set of all possible agents; $\mathcal{S}$ is a set of states; $\mathcal{A}_j$ is agent j's action set; Θ is a set of all possible agent-types. Let the joint action set under a variable agent set $\mathcal{N}_t \subseteq \mathcal{N}$ be defined as that $\mathcal{A}_{\mathcal{N}_t} = \times_{j\in\mathcal{N}_t}\mathcal{A}_j$. Therefore, the joint action space under the variable number of agents is defined as that $\mathcal{A}_{\mathcal{N}} = \bigcup_{\mathcal{N}_t\in\mathbb{P}(\mathcal{N})}\{a|a \in \mathcal{A}_{\mathcal{N}_t}\}$, while the joint agent-type space under the variable number of agents is defined as that $\Theta_{\mathcal{N}} = \bigcup_{\mathcal{N}_t\in\mathbb{P}(\mathcal{N})}\{\theta|\theta \in \Theta^{|\mathcal{N}_t|}\}$. $R : \mathcal{S} \times \mathcal{A}_{\mathcal{N}} \to \mathbb{R}$ is the learner's reward. $T : \mathcal{S} \times \Theta_{\mathcal{N}} \times \mathcal{A}_{\mathcal{N}} \to \mathcal{S} \times \Theta_{\mathcal{N}}$ is a transition function to describe the evolution of states and agents of variable types. The learner's action value function $Q^{\pi^i}(s_t, a_t^i)$ is defined as follows:

$$Q^{\pi^i}(s_t, a_t^i) = \mathbb{E}_{a_t^{-i}\sim\pi_t^{-i}}\left[Q^{\pi^i}(s_t, a_t^{-i}, a_t^i)\right] = \mathbb{E}_{s_t,\theta_t\sim T, a_t^{-i}\sim\pi_t^{-i}, a_t^i\sim\pi^i}\Big[\sum_{t=0}^{\infty}\gamma^t R(s_t, a_t)\Big],$$

where $\gamma \in [0, 1)$ is a discount factor; s_t is a state at timestep t, a_t^{-i} is a joint action of teammates $-i$ at timestep t and a_t^i is the learner i's action at timestep t; π^i is the learner's stationary policy and π_t^{-i} is a joint policy of teammates $-i$; $Q^{\pi^i}(s_t, a_t^{-i}, a_t^i)$ is a joint Q-value. The learner's policy $\pi^{i,*}$ is optimal, if and only if $Q^{\pi^{i,*}}(s_t, a_t^i) \geq Q^{\pi^i}(s_t, a_t^i)$ for all π^i, s_t, a_t^i. The teammates' joint policy is expressed as that $\pi_t^{-i} : \mathcal{S} \times \Theta_{\mathcal{N}} \to \Delta(\mathcal{A}_{\mathcal{N}})$. Although learning an additional learner's policy is not necessary, the definition of teammates' joint policy still implies that the learner's policy can be expressed as that $\pi^i : \mathcal{S} \times \Theta \to \Delta(\mathcal{A}_i)$ for consistency. As a result, any ad hoc team's joint policy (including the learner's policy) can be expressed as that $\pi_t : \mathcal{S} \times \Theta_{\mathcal{N}} \to \Delta(\mathcal{A}_{\mathcal{N}})$ for consistency. The learner is unable to observe the teammates' types and their policies, which can only be inferred through the history states and actions. The learner's decision making is conducted by selecting the actions that maximize $Q^{\pi^i}(s_t, a_t^i)$.

C. Further Details of Implementation

Given the learner's lack of knowledge about P_E and π_t^{-i}, it is essential to discuss strategies for estimating these terms to achieve the convergence of Eq. (7). In the GPL framework, these two terms are implemented as the type inference model and the agent model, respectively. The implementation details are presented below.

C.1. GPL Framework

We now review the GPL's empirical framework (Rahman et al., 2021). This framework consists of the following modules: the type inference model, the joint action value model and the agent model. We only summarize the model specifications. Note that while the original GPL framework is oriented towards a fixed coordination graph, specifically a complete graph, we relax this constraint to accommodate any graph structures as needed.

Type Inference Model. This is modelled as a LSTM (Hochreiter & Schmidhuber, 1997) to infer agent-types of a team at timestep t given that of a team at timestep $t-1$. The agent-type is modelled as a fixed-length hidden-state vector of LSTM, named as agent-type embedding. At each timestep t, the state information of an emergent team $\mathcal{N}_t$ is reproduced to a batch of agents' information $B_t = [\langle u_t, x_{t,1}\rangle, ..., \langle u_t, x_{t,|\mathcal{N}_t|}\rangle]^\top$, where each agent is preserved a vector composing u_t and $x_{t,i}$ which are observations and agent specific information extracted from state s_t. Along with additional information such as the agent-type embedding of $\mathcal{N}_{t-1}$ and the cell state, LSTM estimates the agent-type embedding of $\mathcal{N}_t$. To address the situation of changing team size, at each timestep the agent-type embedding of the agents who leave a team would be removed, while the new added agents' agent-type embedding would be set to a zero vector.

Joint Action Value Model. The joint Q-value, denoted as $\hat{Q}^{\pi^i}(s_t, a_t)$, is approximated as the sum of the corresponding

individual utilities, $\hat{Q}_j^{\pi^i}(a_t^j|s_t)$, and pairwise utilities, $\hat{Q}_{jk}^{\pi^i}(a_t^j, a_t^k|s_t)$, based on the coordination graph structure. The approximation is expressed as follows:

$$\hat{Q}^{\pi^i}(s_t, a_t) = \sum_{j\in\mathcal{N}_t} \hat{Q}_j^{\pi^i}(a_t^j|s_t) + \sum_{(j,k)\in\mathcal{E}_t} \hat{Q}_{jk}^{\pi^i}(a_t^j, a_t^k|s_t).$$

Both $\hat{Q}_j^{\pi^i}(a_t^j|s_t)$ and $\hat{Q}_{jk}^{\pi^i}(a_t^j, a_t^k|s_t)$ are implemented as multilayer perceptrons (MLPs) parameterised by β and δ, denoted as MLP_β and MLP_δ. The input of MLP_β is the concatenation of the learner's agent-type embedding θ_t^i and the teammate j's agent-type embedding θ_t^j. Its output is a vector with a length of $|\mathcal{A}_j|$ estimating $Q_j^{\pi^i}(a_t^j|s_t)$. The detailed expression is shown as follows:

$$\hat{Q}_j^{\pi^i}(a_t^j|s_t) = \text{MLP}_\beta(\theta_t^j, \theta_t^i)(a_t^j).$$

The pairwise utility $\hat{Q}_{jk}^{\pi^i}(a_t^j, a_t^k|s_t)$ is approximated by low-rank factorization, as follows:

$$\hat{Q}_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) = \big(\text{MLP}_\delta(\theta_t^j, \theta_t^i)^\top \text{MLP}_\delta(\theta_t^k, \theta_t^i)\big)(a_t^j, a_t^k),$$

where the input of MLP_δ is the same as MLP_β; the output of $\text{MLP}_\delta(\theta_t^j, \theta_t^i)$ is a matrix with the shape $K \times |\mathcal{A}_j|$ and $K \ll |\mathcal{A}_j|$.

Agent Model. It is assumed that all other connected agents, as described by a coordination graph, would influence an agent's actions. To model this situation, GNN is applied to process the agent-type embedding of a temporary team, denoted as θ_t, where each team member is represented as a node. More specifically, a GNN model called relational forward model (RFM) (Tacchetti et al., 2019) parameterised by η is applied to transform θ_t (as the initial node representation) to $\bar{n}_t$ (as the new node representation) considering other agents' effects. Then, $\bar{n}_t$ is employed to infer $q_{\zeta,\eta}(a_t^{-i}|s_t)$, as the approximation of teammates' joint policy, $\pi_t^{-i}(a_t^{-i}|s_t, \theta_t^{-i})$. The detailed expression is as follows:

$$q_{\zeta,\eta}(a_t^{-i}|s_t) = \prod_{j\in -i} q_{\zeta,\eta}(a_t^j|s_t),$$
$$q_{\zeta,\eta}(a_t^j|s_t) = \text{Softmax}(\text{MLP}_\eta(\bar{n}_t^j))(a_t^j).$$

Learner's Action Value Model. Substituting the agent model and the joint action value model defined above into Eq. (2), the learner's Q-value for its own decision making is approximated as follows:

$$\begin{aligned}\hat{Q}^{\pi_i}(s_t, a_t^i) = \hat{Q}_i^{\pi^i}(a_t^i|s_t) &+ \sum_{a_t^j\in\mathcal{A}_j,(j,i)\in\mathcal{E}_t} \Big(\hat{Q}_j^{\pi^i}(a_t^j|s_t) + \hat{Q}_{ij}^{\pi^i}(a_t^i, a_t^j|s_t)\Big) q_{\zeta,\eta}(a_t^j|s_t) \\ &+ \sum_{\substack{a_t^j\in\mathcal{A}_j, a_t^k\in\mathcal{A}_k,\\ (j,k)\in\mathcal{E}_t}} \hat{Q}_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) q_{\zeta,\eta}(a_t^j|s_t) q_{\zeta,\eta}(a_t^j|s_t).\end{aligned}$$

C.2. Overall Training Procedure of CIAO

We now summarize the overall training procedure of CIAO in Algorithm 1. Note that in the GPL framework, the type inference model is absorbed into the joint Q-value and the agent model as a LSTM, respectively. This construction aims to prevent these two models' gradients from interfering against each other during training (Rahman et al., 2021).

D. Assumptions

In addition to the assumptions of ad hoc teamwork described in the survey (Mirsky et al., 2022), we also list the necessary assumptions for this paper as follow.

Assumption 1. *The following conditional independencies are assumed to hold in any distribution P over the set of variables in an OSB-CAG: (1) $(\theta_t \perp\!\!\!\perp \theta_{t-1}, s_{t-1}, a_{t-1} \mid \mathcal{N}_t, s_t)$; (2) $(\mathcal{N}_t, s_t \perp\!\!\!\perp \theta_{t-1} \mid \mathcal{N}_{t-1}, s_{t-1}, a_{t-1})$; (3) $(\mathcal{N}_t \perp\!\!\!\perp a_t|s_t, \theta_t)$; (4) $(\theta_t^j \perp\!\!\!\perp -j, \theta_t^{-j} \mid \{j\}, s_t)$.*

Algorithm 1 Overall training procedure of CIAO

Input: dynamic affinity graph structure G, number of training episodes e, length of an episode T, replay buffer $\mathcal{B}$
repeat
  Clear the replay buffer $\mathcal{B}$.
  Reset the environment and receive the initial observations.
  for timestep = 1 **to** T **do**
    Execute learner's action by ϵ-greedy policy.
    Store observations (including teammates' actions) for the current timestep in the replay buffer $\mathcal{B}$.
  end for
  Generate the joint Q-value and the agent model as per GPL framework, based on the dynamic affinity graph G.
  Update parameters of pairwise utilities and individual utilities by the loss function proposed in Section 4.3.
  Update parameters of the agent model by the following loss function:

$$L(\zeta, \eta) = -\frac{1}{T}\sum_{t=1}^{T} \log q_{\zeta,\eta}(a_t^{-i}|s_t).$$

until meeting the number of training episodes m

Assumption 1 indicates the assumptions encoding the relationships among random variables that are entailed by any probability distribution describing the *open ad hoc teamwork* process, referred to as conditional independencies (Koller & Friedman, 2009, Ch. 2).

As for conditional independence (1), it implies that the agent-types θ_t for the current timestep are conditionally independent of the related variables $\theta_{t-1}, s_{t-1}, a_{t-1}$ for the preceding timestep, given the agent set $\mathcal{N}_t$ and the state s_t for the current timestep. This is reflected by $P_E(\theta_t|\mathcal{N}_t, s_t) = P(\theta_t|\mathcal{N}_t, s_t, s_{t-1}, a_{t-1}, \theta_{t-1})$.

As for conditional independence (2), it implies that the agent set $\mathcal{N}_t$ and the state s_t for the current timestep is independent of the agent-types θ_{t-1} for the preceding timestep, given the variables $\mathcal{N}_{t-1}, s_{t-1}, a_{t-1}$ for the preceding timestep. This is reflected by $P_T(\mathcal{N}_t, s_t|\mathcal{N}_{t-1}, s_{t-1}, a_{t-1}) = P(\mathcal{N}_t, s_t|\mathcal{N}_{t-1}, s_{t-1}, a_{t-1}, \theta_{t-1})$.

As for conditional independence (3), it implies that the agent set $\mathcal{N}_t$ is independent of the joint action a_t, given the state s_t and the agent-type set θ_t for the same timestep. This is reflected by $P(\mathcal{N}_t|s_t, \theta_t) = P(\mathcal{N}_t|s_t, a_t, \theta_t)$. Note that this condition coincides with scenarios encoded by Assumption 3, where the agent j's policy is able to be varied across timesteps, and the policy is only correlated with its agent-type. In turn, this implies that an agent's mind could be changed across timesteps, which is an evidence that open ad hoc teamwork is also suitable for modelling human-AI cooperation (Shneiderman, 2020; De Peuter & Kaski, 2023). However, for clarity and simplicity to introduce our theory, we assume in this paper that the policy is fixed (time invariant or stationary) across timesteps, as shown in Assumption 6.

As for conditional independence (4), it implies that an agent j's agent-type θ_t^j for some timestep is conditionally independent of other agents $-j$ and their agent-types θ_t^{-j}, given itself denoted as j and the state s_t for that timestep. This is reflected by $\prod_{j=1}^{|\mathcal{N}_t|} P_A(\theta_t^j|\{j\}, s_t) = P_E(\theta_t|\mathcal{N}_t, s_t)$.

Assumption 2. *Suppose that $\alpha_{jk}(a_t^j, a_t^k|s_t) = 0$ for $t \geq T$, where T is the timestep when agent j or k leaves the environment, and $R_j(a_t^j|s_t) = 0$ for $t \geq T'$, where T' is the timestep when agent j leaves the environment.*

Assumption 2 introduces a metric to quantify the impact of agents leaving the environment. Essentially, it posits that an agent that has departed from the environment no longer exerts any influence on the remaining agents within the environment.

Assumption 3. *There exists an underlying agent-type set to generate ad hoc teammates in an environment which is unknown to the learner.*

Assumption 3 provides a natural framework for describing the agent-types of teammates. In scenarios where the agent-type set is sufficiently large, traversing all possible agent-types or compositions becomes impractical. Therefore, this assumption ensures that the generalizability of open ad hoc teamwork is not compromised.

Assumption 4. *Teammates can be influenced by the learner through its decision making.*

Assumption 4 constitutes a fundamental and commonly assumed property essential for rationalizing the ad hoc teamwork

problem. Often referred to as the reactivity of teammates (Barrett et al., 2017), this assumption posits that teammates must be capable of reacting to or being influenced by the learner. Without this interaction, the problem would regress to a scenario akin to a single-agent problem, where teammates merely function as moving 'obstacles.' To avert such a pathological situation, maintaining this assumption serves as a crucial boundary for ad hoc teamwork.

Assumption 5. *The agents stay in the environment at least for a period of timesteps.*

Assumption 5 is a prerequisite ensuring the feasibility of completing arbitrary tasks. Without this condition, wherein an agent joining at a given timestep remains in the environment for a non-instantaneous duration, there would be minimal opportunity for teams of agents to react to and influence each other effectively.

Assumption 6. *Each teammate of an arbitrary agent-type is equipped with a fixed policy.*

Assumption 6 serves as a simplified condition for analyzing the learner's convergence to the optimal policy. By assuming fixed policies for teammates, the Markov process becomes stationary from the learner's perspective, facilitating a more tractable analysis of convergence dynamics. However, this can be further relaxed to cater for more realistic situations.

Assumption 7. *Graph neural networks are assumed to enjoy the generalizability to agent-types.*

Assumption 7 serves as a justification to explain why graph neural networks as a critical component can result in better performance for GPL as reported in the previous work (Rahman et al., 2021) and therefore CIAO (which is implemented upon GPL) in this paper, in generalization of agent-type sets. This assumption can be further relaxed through investigating the underlying reasons behind this phenomenon.

E. Generalization of Preference Values for Coalitional Affinity Game

At the beginning, it is worth noting that in the original work of CAG (Brânzei & Larson, 2009), the definition of the preference value of an arbitrary agent j is as follows:

$$\bar{v}_j(\mathcal{C}) = \begin{cases} 0 & \text{if } \mathcal{C} = \{j\}, \\ \sum_{(j,k)\in\mathcal{E},k\in\mathcal{C}} \bar{w}(j,k) & \text{otherwise.} \end{cases} \tag{9}$$

While the condition that each agent's preference value of a coalition including only itself, equal to zero, is convenient and straightforward for analysis, it imposes limitations on the representational capacity for a wide variety of situations. To address this issue, we generalize the definition of the preference value function in Eq. (9) to the form as follows:

$$v_j(\mathcal{C}) = \begin{cases} b_j \geq 0 & \text{if } \mathcal{C} = \{j\}, \\ \sum_{(j,k)\in\mathcal{E},k\in\mathcal{C}} w(j,k) & \text{otherwise.} \end{cases} \tag{10}$$

The main difference between the definitions in Eq. (9) and Eq. (10) is that the preference value of the coalition only including a single agent is not forced to be zero in Eq. (10). Albeit that the results shown in the original work of CAG (Brânzei & Larson, 2009) are based on each agent's original preference value function shown in Eq. (9), we can still generalize and leverage the results through conducting translation to each agent's preference value function by its preference value of the coalition including itself, $v_j(\{j\}) = b_j \geq 0$ (where b_j is a constant), to align with the condition of $\bar{v}_j(\mathcal{C})$ in Eq. (9). In more details, we can transform the newly defined preference value function in Eq. (10) as follows:

$$\hat{v}_j(\mathcal{C}) = v_j(\mathcal{C}) - v_j(\{j\}) = \begin{cases} 0 & \text{if } \mathcal{C} = \{j\}, \\ \sum_{(j,k)\in\mathcal{E},k\in\mathcal{C}} w(j,k) - v_j(\{j\}) & \text{otherwise.} \end{cases} \tag{11}$$

Therefore, we can directly leverage the results from the previous work (Brânzei & Larson, 2009) by replacing $\bar{v}_j(\mathcal{C})$ with $\hat{v}_j(\mathcal{C})$, and generalize the results to the newly defined preference value function in Eq. (10) by conducting the change of variables according to Eq. (11).

The generalised preference value $v_j(\mathcal{C})$ plays an important role of proving the results of OSB-CAG in the following sections.

Definition 4. *In a CAG with the generalised preference value function, for any agent j, its preference value of the coalition including only itself is defined as that $v_j(\{j\}) \geq 0$.*

In the conventional definition of a coalition value function[5] in the cooperative game theory, the value of an empty set (empty coalition) is defined as zero (Chalkiadakis et al., 2022). In the context of a CAG, we can formally extend the domain of an agent j's preference value function by considering the empty set such that $v_j(\varnothing) = 0$. This extension can be interpreted as that an agent imagines a scenario where it is not included (with no incentives to join). If $v_j(\{j\}) < 0$, it may lead to a paradox that an agent j would choose to disappear from the environment (e.g. suicide) to escape independence owing to $v_j(\{j\}) < v_j(\varnothing)$, which is apparently opposite to morality and ethics. To avoid the paradox, it is reasonable to generalise an agent j's preference value of the coalition including itself to only the non-negative range such that $v_j(\{j\}) \geq 0$.

F. Derivation of Definition 2

Definition 5. *We say that a blocking coalition $\mathcal{C}$ weakly blocks a coalition structure $\mathcal{CS}$ if every agent $j \in \mathcal{C}$ weakly prefers $\mathcal{C}$ to $\mathcal{CS}(j)$ and there exists at least one agent $k \in \mathcal{C}$ who strictly prefers $\mathcal{C}$ to $\mathcal{CS}(j)$. A coalition structure $\mathcal{CS} = \{\mathcal{C}_1, ..., \mathcal{C}_m\}$ admitting no weakly blocking coalition $\mathcal{C} \subset \mathcal{C}_k$, for some $1 \leq k \leq m$, is called inner stable.*

Theorem 4 (Brânzei & Larson (2009))**.** *If a CAG is symmetric, then the social-welfare maximizing partition exhibits inner stability.*

Theorem 4 directly holds for the newly defined $v_j(\mathcal{C})$ in this paper, since it is irrelevant to the detailed representation (feasible domain) of a preference value function (see Theorem 2 and 5 in the previous work (Brânzei & Larson, 2009)).

Lemma 1. *If a CAG is symmetric, then maximizing the social welfare under a grand coalition results in strict core stability.*

Proof. Following Definition 5, it is not difficult to observe that a grand coalition exhibiting strict core stability is equivalent to a grand coalition exhibiting inner stability. Therefore, we can directly obtain the result by Theorem 4. □

F.1. Derivation of Dynamic Variational Strict Core

In an OSB-CAG, at any timestep t, under an arbitrary state $s_t \in \mathcal{S}$ along with a temporary team (including the learner i), denoted as $\mathcal{N}_t \subseteq \mathcal{N}$, and the temporary team's joint action $a_t \in \mathcal{A}_{\mathcal{N}_t}$, the coalition reward can be equivalently expressed as a preference value of an agent belonging to a temporary team $\mathcal{N}_t$ such that $R_j(s_t, a_t) = v_j(\mathcal{N}_t)$. The temporary team $\mathcal{N}_t$ can be interpreted as the grand coalition at any timestep t. Thereby, *reaching the strict core stability at any timestep t is equivalent to maximizing the social welfare at the timestep*. Different from the previous work (Brânzei & Larson, 2009) that given the predetermined preference values, the coalition structure is as a decision variable to reach the strict core; in this paper, we predetermine a temporary team, as the target coalition structure, and the learner i's action is as an extended decision variable to change the preference values (coalition rewards) in order to reach the *variational strict core* (VSC) that is defined with the same criterion as the strict core, but with different target variables as elements to form the solution set. The learner i's action is generated by its policy π^i. By Assumption 4, we can get that the learner's action is able to influence teammates' actions. Therefore, the teammates' coalition rewards as an evaluation of their policies will also be varied accordingly. This explains that the learner's action can be seen as a decision variable that is able to indirectly change teammates' coalition rewards. By Lemma 1, if a dynamic affinity graph at timestep t is symmetric, we can express the VSC for any timestep t (under an arbitrary $s_t \in \mathcal{S}$ along with a temporary team $\mathcal{N}_t \subseteq \mathcal{N}$, a joint agent-type $\theta_t \in \Theta^{|\mathcal{N}_t|}$, the teammates' policies $\pi_t^{-i}(a_t^{-i}|s_t, \theta_t^{-i})$ with respect to their agent-types and the state) to find the learner's optimal action (rather than find a coalition structure in the previous work) as follows:

$$\texttt{VSC} := \Big\{ a^{i,*} \;\Big|\; \sum_{j \in \mathcal{N}_t} R_j(s_t, a_t^{i,*}, a_t^{-i}) \geq \sum_{j \in \mathcal{N}_t} R_j(s_t, a_t^i, a_t^{-i}),\ \forall a_t^i \in \mathcal{A}_i \Big\}. \tag{12}$$

Note that the strict core defined in Eq. (12) implicitly assumes that the teammates' reaction is instantaneous (happening at the same timestep). Recall that our aim is to find the learner's optimal stationary policy $\pi^{i,*}$ that generates actions across timesteps (in a long horizon), in order to influence the temporary teammates occurring at any timestep to collaborate (meeting the strict core stability). We now generalize the VSC defined in Eq. (12) by considering the process of generating states, teammates, agent-types and teammates' actions, named as *dynamic variational strict core* (DVSC). The DSVC is defined as follows:

$$\texttt{DVSC} := \Big\{ \pi^{i,*} \;\Big|\; \mathbb{E}_{\pi^{i,*}}\Big[\sum_{t=0}^{\infty} \gamma^t \sum_{j \in \mathcal{N}_t} R_j(s_t, a_t)\Big] \geq \mathbb{E}_{\pi^i}\Big[\sum_{t=0}^{\infty} \gamma^t \sum_{j \in \mathcal{N}_t} R_j(s_t, a_t)\Big], \forall s_0 \in \mathcal{S}, \forall \pi^i \Big\}, \tag{13}$$

[5]The preference value function of an agent can be seen as a coalition value function specifically defined for the agent.

where $a_t^i \sim \pi^i$ and $a_t^{-i} \sim \pi_t^{-i}$; $\mathbb{E}_{\pi^i}[\cdot]$ denotes the expectation that also implicitly depends on $\theta_t \sim P_E, \mathcal{N}_t, s_t \sim P_O$.

Note that the VSC defined in Eq. (13) weakens the implicit assumption of the strict core defined in Eq. (12). In more details, it allows the teammates to react at the successor timesteps instead of the mandatory instantaneous reaction at the same timestep. Nevertheless, this requires that the learner has potential for adapting to the teammates (through interaction with teammates for a period). By Assumption 5 and 6, the learner's adaption to the temporary teammates is possible.

G. Mathematical Proofs

G.1. The Proof of Proposition 1

Proposition 1. *$T(\mathcal{N}_t, s_t, \theta_t | s_{t-1}, a_{t-1}, \theta_{t-1})$ for $t \geq 1$ can be expressed in terms of the following well-defined probability distributions: $P_I(\mathcal{N}_0, s_0)$, $P_T(\mathcal{N}_t, s_t | \mathcal{N}_{t-1}, s_{t-1}, a_{t-1})$ for $t \geq 1$, and $P_A(\theta_t^j | \{j\}, s_t)$ for $t \geq 0$.*

Proof. To ease the proof, we assume that s_t and a_t are discrete variables with no loss of generality. We prove that $T(\mathcal{N}_t, s_t, \theta_t | s_{t-1}, a_{t-1}, \theta_{t-1})$ can be expressed as the probability distributions we have defined, as follows:

$$
\begin{aligned}
& T(\mathcal{N}_t, s_t, \theta_t | s_{t-1}, a_{t-1}, \theta_{t-1}) \\
&= P(\theta_t | \mathcal{N}_t, s_t, s_{t-1}, a_{t-1}, \theta_{t-1}) P_O(\mathcal{N}_t, s_t | s_{t-1}, a_{t-1}, \theta_{t-1}) \\
&= P_E(\theta_t | \mathcal{N}_t, s_t) P_O(\mathcal{N}_t, s_t | s_{t-1}, a_{t-1}, \theta_{t-1}) \quad \text{(By conditional independence (1) in Assumption 1.)} \\
&= P_E(\theta_t | \mathcal{N}_t, s_t) \sum_{\mathcal{N}_{t-1}} P(\mathcal{N}_t, s_t, \mathcal{N}_{t-1} | s_{t-1}, a_{t-1}, \theta_{t-1}) \\
&= P_E(\theta_t | \mathcal{N}_t, s_t) \sum_{\mathcal{N}_{t-1}} P(\mathcal{N}_t, s_t | \mathcal{N}_{t-1}, s_{t-1}, a_{t-1}, \theta_{t-1}) P(\mathcal{N}_{t-1} | s_{t-1}, a_{t-1}, \theta_{t-1}) \\
&= P_E(\theta_t | \mathcal{N}_t, s_t) \sum_{\mathcal{N}_{t-1}} P_T(\mathcal{N}_t, s_t | \mathcal{N}_{t-1}, s_{t-1}, a_{t-1}) P(\mathcal{N}_{t-1} | s_{t-1}, \theta_{t-1}) \\
& \quad \text{(By conditional independence (2) and (3) in Assumption 1.)} \\
&= \prod_{j=1}^{|\mathcal{N}_t|} P_A(\theta_t^j | \{j\}, s_t) \sum_{\mathcal{N}_{t-1}} P_T(\mathcal{N}_t, s_t | \mathcal{N}_{t-1}, s_{t-1}, a_{t-1}) P(\mathcal{N}_{t-1} | s_{t-1}, \theta_{t-1}). \\
& \quad \text{(By conditional independence (4) in Assumption 1.)}
\end{aligned}
$$

To complete the above proof, we need to further show the expression of $P(\mathcal{N}_t | s_t, \theta_t)$ as follows:

$$
P(\mathcal{N}_t | s_t, \theta_t) = \frac{\sum_{s_t} P_E(\theta_t | \mathcal{N}_t, s_t) P(\mathcal{N}_t, s_t)}{\sum_{\mathcal{N}_t} \sum_{s_t} P_E(\theta_t | \mathcal{N}_t, s_t) P(\mathcal{N}_t, s_t)}. \tag{14}
$$

Apparently, we require to prove that $P(\mathcal{N}_t, s_t)$ admits factorization into the probability distributions we have defined. We now conduct this by mathematical induction as follows:

Base case: As per the definition, $P_I(\mathcal{N}_0, s_0)$ is a predefined probability distribution to express $P(\mathcal{N}_0, s_0)$ for $t = 0$.

Induction case: Assume the induction hypothesis that $P(\mathcal{N}_t, s_t)$ admits factorization into the probability distributions we have defined, for any $t \geq 0$.

Next, we aim to prove that $P(\mathcal{N}_{t+1}, s_{t+1})$ admits factorization into the probability distributions we have defined and $P(\mathcal{N}_t, s_t)$ as the induction hypothesis, such that

$$
P(\mathcal{N}_{t+1}, s_{t+1}) = \sum_{\mathcal{N}_t} \sum_{s_t} \sum_{a_t} \sum_{\theta_t} P(\mathcal{N}_{t+1}, s_{t+1}, \mathcal{N}_t, s_t, a_t, \theta_t),
$$

where

$$
\begin{aligned}
P(\mathcal{N}_{t+1}, s_{t+1}, \mathcal{N}_t, s_t, a_t, \theta_t) &= P(\mathcal{N}_{t+1}, s_{t+1} | \mathcal{N}_t, s_t, a_t, \theta_t) P(\mathcal{N}_t | s_t, a_t, \theta_t) P(a_t | s_t, \theta_t) P(s_t, \theta_t) \\
&= P_T(\mathcal{N}_{t+1}, s_{t+1} | \mathcal{N}_t, s_t, a_t) P(\mathcal{N}_t | s_t, \theta_t) P(a_t | s_t, \theta_t) P(s_t, \theta_t) \\
& \quad \text{(By conditional independence (2) and (3) in Assumption 1.)} \\
&= P_T(\mathcal{N}_{t+1}, s_{t+1} | \mathcal{N}_t, s_t, a_t) P(\mathcal{N}_t | s_t, \theta_t) \pi_t(a_t | s_t, \theta_t) P(s_t, \theta_t). \\
& \quad \text{(By the definition of } \pi_t \text{ in Appendix B, we can use } \pi_t \text{ to specify } P(a_t | s_t, \theta_t).)
\end{aligned}
$$

Since $P(\mathcal{N}_t, s_t)$ is the induction hypothesis and $P_E(\theta_t|\mathcal{N}_t, s_t)$ is a probability distribution we have defined, we can derive $P(\mathcal{N}_t|s_t, \theta_t)$ by Eq. (14). Also, by that $P(\mathcal{N}_t, s_t)$ is the induction hypothesis and $P_E(\theta_t|\mathcal{N}_t, s_t)$ is a probability distribution we have defined, we can obtain $P(s_t, \theta_t)$ as follows:

$$\begin{aligned} P(s_t, \theta_t) &= \sum_{\mathcal{N}_t} P(\mathcal{N}_t, s_t, \theta_t) \\ &= \sum_{\mathcal{N}_t} P_E(\theta_t|\mathcal{N}_t, s_t) P(\mathcal{N}_t, s_t). \end{aligned}$$

Conclusion: Since both the base case and the induction step have been proved as true, $P(\mathcal{N}_t, s_t)$ is proved to admit factorization into the probability distributions we have defined for any $t \geq 0$. □

G.2. The Proof of Theorem 1

Theorem 5 (Brânzei & Larson (2009))**.** *In a CAG with an affinity graph $G = \langle \mathcal{N}, \mathcal{E} \rangle$, if for all $(j,k) \in \mathcal{E}$, $\bar{w}(j,k) \geq 0$, then the grand coalition is in the strict core.*

Lemma 2. *In a CAG with an affinity graph $G = \langle \mathcal{N}, \mathcal{E} \rangle$ and the generalised preference value function $v_j(\mathcal{C})$, if the following conditions are satisfied such that*

$$\begin{aligned} w(j,k) &\geq z_{jk}(\{j\}), \\ v_j(\{j\}) &= \sum_{(j,k)\in\mathcal{E}, k\in\mathcal{N}} z_{jk}(\{j\}), \\ &\forall (j,k) \in \mathcal{E}, \end{aligned} \tag{15}$$

then the grand coalition is in the strict core.

Proof. Recall that we have generalised the preference value function in this paper (see Appendix E). Theorem 5 only holds for the case where the preference value function is defined as $\bar{v}_j(\mathcal{C})$ in Eq. (9). As a result, we first investigate the conditions that makes Theorem 5 still hold for the generalised preference value function $v_j(\mathcal{C})$ in Eq. (10). As discussed before, we can transform the generalised preference value function $v_j(\mathcal{C})$ to the feasible domain of the original preference value function $\bar{v}_j(\mathcal{C})$ by translation such that

$$\hat{v}_j(\mathcal{C}) = v_j(\mathcal{C}) - v_j(\{j\}) = \begin{cases} 0 & \text{if } \mathcal{C} = \{j\}, \\ \sum_{(j,k)\in\mathcal{E}, k\in\mathcal{C}} w(j,k) - v_j(\{j\}) & \text{otherwise.} \end{cases}$$

It is apparent that the domain of $\hat{v}_j(\mathcal{C})$ is aligned with that of $\bar{v}_j(\mathcal{C})$. Therefore, we can substitute $\hat{v}_j(\mathcal{C})$ for $\bar{v}_j(\mathcal{C})$. Since Theorem 5 only considers the grand coalition, we can temporarily ignore the case of that $\mathcal{C} = \{j\}$. For any $\mathcal{C} \neq \{j\}$ of $\hat{v}_j(\mathcal{C})$, we can rewrite the expression $\sum_{(j,k)\in\mathcal{E}, k\in\mathcal{C}} w(j,k) - v_j(\{j\})$ as follows:

$$\begin{aligned} \sum_{(j,k)\in\mathcal{E}, k\in\mathcal{C}} w(j,k) - v_j(\{j\}) &= \sum_{(j,k)\in\mathcal{E}, k\in\mathcal{C}} \{w(j,k) - z_{jk}(\{j\})\} \\ &:= \sum_{(j,k)\in\mathcal{E}, k\in\mathcal{C}} \hat{w}(j,k), \end{aligned}$$

where

$$\begin{aligned} \hat{w}(j,k) &:= w(j,k) - z_{jk}(\{j\}), \\ v_j(\{j\}) &= \sum_{(j,k)\in\mathcal{E}, k\in\mathcal{C}} z_{jk}(\{j\}). \end{aligned}$$

By the condition that $\hat{w}(j,k) \geq 0$, for $(j,k) \in \mathcal{E}$, from Theorem 5, we can directly obtain the conditions to enable the grand coalition $\mathcal{N}$ being in the strict core such that

$$\begin{aligned} w(j,k) &\geq z_{jk}(\{j\}), \\ v_j(\{j\}) &= \sum_{(j,k)\in\mathcal{E}, k\in\mathcal{N}} z_{jk}(\{j\}), \\ &\forall (j,k) \in \mathcal{E}. \end{aligned} \tag{16}$$

□

Theorem 1. *In an OSB-CAG, for any dynamic affinity graph $G_t = \langle \mathcal{N}_t, \mathcal{E}_t \rangle$ at any timestep t, if there exists a joint action $a_t \in \mathcal{A}_{\mathcal{N}_t}$, for any agent $j \in \mathcal{N}_t$, satisfying $R_j(a_t|s_t) \geq R_j(a_t^j|s_t)$ for any $s_t \in \mathcal{S}$, then DVSC always exists.*

Proof. To avoid losing the generality, we consider an arbitrary dynamic affinity graph $G_t = \langle \mathcal{N}_t, \mathcal{E}_t \rangle$ for a temporary team $\mathcal{N}_t \subseteq \mathcal{N}$ at an arbitrary timestep t. For any state $s_t \in \mathcal{S}$ and any joint action $a_t \in \mathcal{A}_{\mathcal{N}_t}$, the affinity weight $w_{jk}(a_t^j, a_t^k|s_t)$ of any $(j,k) \in \mathcal{E}_t$ can be represented as a corresponding $w(j,k)$ such that $w(j,k) = w_{jk}(a_t^j, a_t^k|s_t)$. Similarly, each agent j's preference reward for the coalition including only itself $R_j(a_t^j|s_t)$ can also be represented as a corresponding $v_j(\{j\})$ such that $v_j(\{j\}) = R_j(a_t^j|s_t)$. Thereafter, we can apply Lemma 2 to the situation here at a single timestep t. Substituting the above variables into Eq. (15) in Lemma 2, it is not difficult to observe that if for any state $s_t \in \mathcal{S}$, there exists a joint action $a_t \in \mathcal{A}_{\mathcal{N}_t}$ such that $\sum_{(j,k)\in\mathcal{E}_t, k\in\mathcal{N}_t} w_{jk}(a_t^j, a_t^k|s_t) \geq R_j(a_t^j|s_t)$, then there always exists a $R_j(a_t^j|s_t) = \sum_{(j,k)\in\mathcal{E}_t, k\in\mathcal{N}_t} \beta_{jk}(a_t^j|s_t)$ satisfying the condition that $w_{jk}(a_t^j, a_t^k|s_t) \geq \beta_{jk}(a_t^j|s_t)$, for all $(j,k) \in \mathcal{E}_t$, for any state $s_t \in \mathcal{S}$. Analogously, we can obtain the same results for all timesteps as above, which achieves the long-horizon objective as defined in the DVSC. Therefore, we can conclude that for any dynamic affinity graph $G_t = \langle \mathcal{N}_t, \mathcal{E}_t \rangle$ at any timestep t, if there exists a joint action $a_t \in \mathcal{A}_{\mathcal{N}_t}$, for any agent $j \in \mathcal{N}_t$, satisfying $R_j(a_t|s_t) = \sum_{(j,k)\in\mathcal{E}_t, k\in\mathcal{N}_t} w_{jk}(a_t^j, a_t^k|s_t) \geq R_j(a_t^j|s_t)$ for any $s_t \in \mathcal{S}$, then the DVSC defined in Eq. (4) always exists. □

G.3. The Proof of Theorem 2

Lemma 3. *Under Assumption 2, it is valid to have the expressions that $Q_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) = \mathbb{E}_{\pi^i}[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} \alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau)]$ and $Q_j^{\pi^i}(a_t^j|s_t) = \mathbb{E}_{\pi^i}[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} R_j(a_\tau^j|s_\tau)]$, with the learner i's policy π^i.*

Proof. Suppose that agent j or k leaves the environment at timestep T, then we can have the expression that $Q_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) = \mathbb{E}_{\pi^i}[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} \alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau)]$ by the condition in Assumption 2 that $\alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau) = 0$ for $\tau \geq T$ if agent j or k leaves the environment at timestep T as follows:

$$
\begin{aligned}
Q_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) &= \mathbb{E}_{\pi^i}[\sum_{\tau=t}^{T} \gamma^{\tau-t} \alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau)] \\
&= \mathbb{E}_{\pi^i}[\sum_{\tau=t}^{T} \gamma^{\tau-t} \alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau) + \underbrace{\sum_{\tau=T}^{\infty} \gamma^{\tau-t} \alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau)}_{=0 \text{ by Assumption 2.}}] \\
&= \mathbb{E}_{\pi^i}[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} \alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau)].
\end{aligned}
$$

Similarly, by the condition in Assumption 2 that $R_j(a_\tau^j|s_\tau) = 0$ for $\tau \geq T'$ if agent j leaves the environment at timestep T', we can derive the result that $Q_j^{\pi^i}(a_t^j|s_t) = \mathbb{E}_{\pi^i}[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} R_j(a_\tau^j|s_\tau)]$. □

Theorem 2. *Under Assumption 2, if $w_{jk}(a_\tau^j, a_\tau^k|s_\tau) = \alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau) + \beta_{jk}(a_\tau^j|s_\tau)$, then the joint Q-value of the learner's policy π^i can be expressed as follows:*

$$
\begin{aligned}
Q^{\pi^i}(s_t, a_t) &= \sum_{(j,k)\in\mathcal{E}_t} Q_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) + \sum_{j\in\mathcal{N}_t} Q_j^{\pi^i}(a_t^j|s_t) \\
&= \sum_{j\in\mathcal{N}_t} \Big\{ \sum_{(j,k)\in\mathcal{E}_t} Q_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) + \sum_{j\in\mathcal{N}_t} Q_j^{\pi^i}(a_t^j|s_t) \Big\} \\
&:= \sum_{j\in\mathcal{N}_t} Q_j^{\pi^i}(a_t|s_t),
\end{aligned}
$$

where $Q_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) = \mathbb{E}_{\pi^i}[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} \alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau)]$ and $Q_j^{\pi^i}(a_t^j|s_t) = \mathbb{E}_{\pi^i}[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} R_j(a_\tau^j|s_\tau)]$.

Proof. By Assumption 2 and the result of Lemma 3, for any state $s_t \in \mathcal{S}$ and any joint action $a_t \in \mathcal{A}_{\mathcal{N}_t}$, we can represent the

joint Q-value under any learner i's policy π^i such as $Q^{\pi^i}(s_t, a_t)$ as follows:

$$
\begin{aligned}
Q^{\pi^i}(s_t, a_t) &= \mathbb{E}_{\pi^i}\Big[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}R(s_\tau, a_\tau)\Big] \\
&= \mathbb{E}_{\pi^i}\Big[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}\sum_{j\in\mathcal{N}_\tau}R_j(a_\tau|s_\tau)\Big] \\
&= \mathbb{E}_{\pi^i}\Big[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}\sum_{j\in\mathcal{N}_\tau}\Big(\sum_{(j,k)\in\mathcal{E}_\tau}\alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau) + R_j(a_\tau^j|s_\tau)\Big)\Big] \\
&= \mathbb{E}_{\pi^i}\Bigg[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}\Bigg(\sum_{j\in\mathcal{N}_\tau}\Big(\sum_{(j,k)\in\mathcal{E}_\tau}\alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau) + R_j(a_\tau^j|s_\tau)\Big) \\
&\qquad + \underbrace{\sum_{j\in\mathcal{N}_t\setminus\mathcal{N}_\tau}\Big(\sum_{(j,k)\in\mathcal{E}_t\setminus\mathcal{E}_\tau}\alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau) + R_j(a_\tau^j|s_\tau)\Big)}_{=0 \text{ by Assumption 2.}}\Bigg)\Bigg] \\
&= \mathbb{E}_{\pi^i}\Big[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}\sum_{j\in\mathcal{N}_t}\Big(\sum_{(j,k)\in\mathcal{E}_t}\alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau) + R_j(a_\tau^j|s_\tau)\Big)\Big] \\
&= \sum_{j\in\mathcal{N}_t}\Bigg\{\sum_{(j,k)\in\mathcal{E}_t}\underbrace{\mathbb{E}_{\pi^i}\Big[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}\alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau)\Big]}_{=Q_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) \text{ by Lemma 3.}} + \underbrace{\mathbb{E}_{\pi^i}\Big[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}R_j(a_\tau^j|s_\tau)\Big]}_{=Q_j^{\pi^i}(a_t^j|s_t) \text{ by Lemma 3.}}\Bigg\} \\
&= \sum_{j\in\mathcal{N}_t}\Bigg\{\sum_{(j,k)\in\mathcal{E}_t}Q_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) + Q_j^{\pi^i}(a_t^j|s_t)\Bigg\} \\
&= \sum_{j\in\mathcal{N}_t}\sum_{(j,k)\in\mathcal{E}_t}Q_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) + \sum_{j\in\mathcal{N}_t}Q_j^{\pi^i}(a_t^j|s_t) \\
&= \sum_{(j,k)\in\mathcal{E}_t}Q_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) + \sum_{j\in\mathcal{N}_t}Q_j^{\pi^i}(a_t^j|s_t).
\end{aligned}
\tag{17}
$$

By the fashion of Bellman optimality equation, for any state $s_t \in \mathcal{S}$ and any joint action $a_t \in \mathcal{A}_{\mathcal{N}_t}$, we can write out each agent j's preference Q-value under the learner i's policy π^i, $Q_j^{\pi^i}(a_t|s_t)$, as follows:

$$
\begin{aligned}
Q_j^{\pi^i}(s_t, a_t) &= \mathbb{E}_{\pi^i}\Big[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}R_j(a_\tau|s_\tau)\Big] \\
&= \mathbb{E}_{\pi^i}\Big[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}\Big(\sum_{(j,k)\in\mathcal{E}_\tau}\alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau) + R_j(a_\tau^j|s_\tau)\Big)\Big] \\
&= \mathbb{E}_{\pi^i}\Big[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}\Big(\sum_{(j,k)\in\mathcal{E}_\tau}\alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau)\Big)\Big] + \mathbb{E}_{\pi^i}\Big[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}R_j(a_\tau^j|s_\tau)\Big] \\
&= \mathbb{E}_{\pi^i}\Big[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}\Big(\sum_{(j,k)\in\mathcal{E}_\tau}\alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau) + \underbrace{\sum_{(j,k)\in\mathcal{E}_t\setminus\mathcal{E}_\tau}\alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau)}_{=0 \text{ by Assumption 2.}}\Big)\Big] + \mathbb{E}_{\pi^i}\Big[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}R_j(a_\tau^j|s_\tau)\Big] \\
&= \mathbb{E}_{\pi^i}\Big[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}\Big(\sum_{(j,k)\in\mathcal{E}_t}\alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau)\Big)\Big] + \mathbb{E}_{\pi^i}\Big[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}R_j(a_\tau^j|s_\tau)\Big] \\
&= \sum_{(j,k)\in\mathcal{E}_t}\underbrace{\mathbb{E}_{\pi^i}\Big[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}\alpha_{jk}(a_\tau^j, a_\tau^k|s_\tau)\Big]}_{=Q_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) \text{ by Lemma 3.}} + \underbrace{\mathbb{E}_{\pi^i}\Big[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}R_j(a_\tau^j|s_\tau)\Big]}_{=Q_j^{\pi^i}(a_t^j|s_t) \text{ by Lemma 3.}} \\
&= \sum_{(j,k)\in\mathcal{E}_t}Q_{jk}^{\pi^i}(a_t^j, a_t^k|s_t) + Q_j^{\pi^i}(a_t^j|s_t).
\end{aligned}
\tag{18}
$$

By substituting the expression of $Q_j^{\pi^i}(s_t, a_t)$ derived in Eq. (18) into Eq. (17), we can get the following relation:

$$Q^{\pi^i}(s_t, a_t) = \sum_{j \in \mathcal{N}_t} Q_j^{\pi^i}(a_t|s_t). \tag{19}$$

□

G.4. The Proof of the Conditions of Symmetry for Various Dynamic Affinity Graphs

Proposition 3. *For the learner i and any teammate j or k, the constraints $R_i(a_t^i|s_t) = \sum_{j \in -i} R_j(a_t^j|s_t)$ and $\alpha_{jk}(a_t^j, a_t^k|s_t) = \alpha_{kj}(a_t^k, a_t^j|s_t)$, for any $a_t \in \mathcal{A}_{\mathcal{N}_t}$ and $s_t \in \mathcal{S}$, are necessary for a star dynamic affinity graph to be symmetric.*

Proof. Recall that a symmetric dynamic affinity graph $G_t = \langle \mathcal{N}_t, \mathcal{E}_t \rangle$ needs to satisfy the following condition that $w_{jk}(a_t^j, a_t^k|s_t) = w_{kj}(a_t^k, a_t^j|s_t)$, for all $(j,k) \in \mathcal{E}_t$, for any state $s_t \in \mathcal{S}$ and any joint action $a_t \in \mathcal{A}_{\mathcal{N}_t}$. In the dynamic affinity graph as a star graph, the affinity weights of any $(i,j) \in \mathcal{E}_t$ or $(j,i) \in \mathcal{E}_t$ can be represented as follows:

$$w_{ij}(a_t^i, a_t^j|s_t) = \alpha_{ij}(a_t^i, a_t^j|s_t) + \beta_{ij}(a_t^i|s_t), \text{where } R_i(a_t^i|s_t) = \sum_{j \in -i} \beta_{ij}(a_t^i|s_t),$$
$$w_{ji}(a_t^j, a_t^i|s_t) = \alpha_{ji}(a_t^j, a_t^i|s_t) + \beta_{ji}(a_t^j|s_t), \text{where } R_j(a_t^j|s_t) = \beta_{ji}(a_t^j|s_t).$$

It is not difficult to observe that for all $\forall s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}_{\mathcal{N}_t}$ the following conditions that

$$\alpha_{ij}(a_t^i, a_t^j|s_t) = \alpha_{ji}(a_t^j, a_t^i|s_t),$$
$$R_i(a_t^i|s_t) = \sum_{j \in -i} R_j(a_t^j|s_t),$$

are necessary for that the star dynamic affinity graph is symmetric. In more details, that $R_i(a_t^i|s_t) = \sum_{j \in -i} R_j(a_t^j|s_t)$ is a necessary condition for the existence of the one-to-one correspondence that $\beta_{ij}(a_t^i|s_t) = \beta_{ji}(a_t^j|s_t) = R_j(a_t^j|s_t)$. □

Proposition 4. *For any two agents j or k, the constraints $R_j(a_t^j|s_t) = R_k(a_t^k|s_t)$ and $\alpha_{jk}(a_t^j, a_t^k|s_t) = \alpha_{kj}(a_t^k, a_t^j|s_t)$, for any $a_t \in \mathcal{A}_{\mathcal{N}_t}$ and $s_t \in \mathcal{S}$, are necessary for the complete dynamic affinity graph to be symmetric.*

Proof. Recall that a symmetric dynamic affinity graph $G_t = \langle \mathcal{N}_t, \mathcal{E}_t \rangle$ needs to satisfy the following condition that $w_{jk}(a_t^j, a_t^k|s_t) = w_{kj}(a_t^k, a_t^j|s_t)$, for all $(j,k) \in \mathcal{E}_t$, for any state $s_t \in \mathcal{S}$ and any joint action $a_t \in \mathcal{A}_{\mathcal{N}_t}$. In the dynamic affinity graph as a complete graph, the affinity weights of any $(j,k) \in \mathcal{E}_t$ can be represented as follows:

$$w_{jk}(a_t^j, a_t^k|s_t) = \alpha_{jk}(a_t^j, a_t^k|s_t) + \beta_{jk}(a_t^j|s_t), \text{where } R_j(a_t^j|s_t) = \sum_{k \in -j} \beta_{jk}(a_t^j|s_t).$$

It is not difficult to observe that for all $\forall s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}_{\mathcal{N}_t}$ the following conditions that

$$\alpha_{jk}(a_t^j, a_t^k|s_t) = \alpha_{kj}(a_t^k, a_t^j|s_t),$$
$$R_j(a_t^j|s_t) = R_k(a_t^k|s_t),$$

are necessary for that the complete dynamic affinity graph is symmetric. In more details, that $R_j(a_t^j|s_t) = \sum_{k \in -j} \beta_{jk}(a_t^j|s_t) = \sum_{j \in -k} \beta_{kj}(a_t^k|s_t) = R_k(a_t^k|s_t)$ is a necessary condition for the existence of the one-to-one correspondence that $\beta_{jk}(a_t^j|s_t) = \beta_{kj}(a_t^k|s_t)$. □

G.5. The Proof of Theorem 3

Theorem 3. *Under Assumption 2 and an arbitrary learner's deterministic stationary policy π^i, the Bellman equation for the OSB-CAG with DVSC as a solution concept is expressed as follows: $Q^{\pi^i}(s_t, a_t) = R(s_t, a_t) + \gamma \mathbb{E}_{\mathcal{N}_{t+1}, s_{t+1} \sim P_O}\Big[\mathbb{E}_{\theta_{t+1} \sim P_E,\ a_{t+1} \sim \pi_{t+1}}\big[Q^{\pi^i}(s_{t+1}, a_{t+1})\big]\Big]$.*

Proof. We derive Eq. (6) as follows.

By the result of Theorem 2, we can represent the joint Q-value under an arbitrary learner's deterministic stationary policy π^i referred to as $Q^{\pi^i}(s_t, a_t)$ as follows:

$$Q^{\pi^i}(s_t, a_t) = \sum_{j \in \mathcal{N}_t} Q_j^{\pi^i}(a_t|s_t), \tag{20}$$

Next, we can expand the preference Q-value of each agent $j \in \mathcal{N}_t$ following the fashion of the Bellman equation such that

$$Q_j^{\pi^i}(a_t|s_t) = R_j(a_t|s_t) + \gamma \mathbb{E}_{\mathcal{N}_{t+1}, s_{t+1} \sim P_O}\Big[\mathbb{E}_{\substack{\theta_{t+1} \sim P_E, \\ a_{t+1} \sim \pi_{t+1}}}\big[Q_j^{\pi^i}(a_{t+1}|s_{t+1})\big]\Big]. \tag{21}$$

Then, we can sum up Eq. (21) for all possible agents belonging to the temporary team $\mathcal{N}_t$ and get an equation to evaluate the influence of the learner's policy π^i to a temporary team $\mathcal{N}_t$ such that

$$\begin{aligned}
Q^{\pi^i}(s_t, a_t) &= \sum_{j \in \mathcal{N}_t} Q_j^{\pi^i}(a_t|s_t) \\
&= \sum_{j \in \mathcal{N}_t} R_j(a_t|s_t) + \sum_{j \in \mathcal{N}_t} \gamma \mathbb{E}_{\mathcal{N}_{t+1}, s_{t+1} \sim P_O}\Big[\mathbb{E}_{\substack{\theta_{t+1} \sim P_E, \\ a_{t+1} \sim \pi_{t+1}}}\big[Q_j^{\pi^i}(a_{t+1}|s_{t+1})\big]\Big] \\
&= R(s_t, a_t) + \underbrace{\sum_{j \in \mathcal{N}_{t+1}} \gamma \mathbb{E}_{\mathcal{N}_{t+1}, s_{t+1} \sim P_O}\Big[\mathbb{E}_{\substack{\theta_{t+1} \sim P_E, \\ a_{t+1} \sim \pi_{t+1}}}\big[Q_j^{\pi^i}(a_{t+1}|s_{t+1})\big]\Big]}_{\text{Since } Q_j^{\pi^i}(a_{t+1}|s_{t+1}) = 0 \text{ for agent } j \in \mathcal{N}_t \setminus \mathcal{N}_{t+1} \text{ by Assumption 2.}} \\
&= R(s_t, a_t) + \gamma \mathbb{E}_{\mathcal{N}_{t+1}, s_{t+1} \sim P_O}\Big[\mathbb{E}_{\substack{\theta_{t+1} \sim P_E, \\ a_{t+1} \sim \pi_{t+1}}}\big[\sum_{j \in \mathcal{N}_{t+1}} Q_j^{\pi^i}(a_{t+1}|s_{t+1})\big]\Big] \\
&= R(s_t, a_t) + \gamma \mathbb{E}_{\mathcal{N}_{t+1}, s_{t+1} \sim P_O}\Big[\mathbb{E}_{\substack{\theta_{t+1} \sim P_E, \\ a_{t+1} \sim \pi_{t+1}}}\big[Q^{\pi^i}(s_{t+1}, a_{t+1})\big]\Big].
\end{aligned} \tag{22}$$

Note that Eq. (22) does not hold if $\mathcal{N}_t \subset \mathcal{N}_{t+1}$, since it is problematic to expand the preference Q-value of an agent $k \in \mathcal{N}_{t+1}$ but $\notin \mathcal{N}_t$ at timestep t, which can be seen as a singularity of this equation. More specifically, $0 = Q_k^{\pi^i}(a_t|s_t) = R_k(a_t|s_t) + \gamma \mathbb{E}_{\mathcal{N}_{t+1}, s_{t+1} \sim P_O}\Big[\mathbb{E}_{\substack{\theta_{t+1} \sim P_E, \\ a_{t+1} \sim \pi_{t+1}}}\big[Q_k^{\pi^i}(a_{t+1}|s_{t+1})\big]\Big] > 0$ is impossible, given that at least $R_k(a_{t'}|s_{t'}) > 0$, implying agent k's preference for collaborating with other agents, at a timestep $t' \geq t$. □

H. Experimental Settings

We evaluate our proposed CIAO in two existing environments, LBF and Wolfpack, both configured with open team settings (Rahman et al., 2021). In these settings, teammates are randomly selected to enter the environment and remain for a specified number of timesteps. If a teammate surpasses its allocated lifetime, it is removed from the environment and placed in a re-entry queue with a randomly assigned waiting time. The randomized re-entry queue results in varied compositions of teammates in a temporary team. When the number of agents in the environment does not reach its maximum, agents in the re-entry queue are introduced to the environment. Specifically, in the Wolfpack environment, we uniformly determine the active duration by selecting a value between 25 and 35 timesteps, while the dead duration is uniformly sampled between 15 and 25 timesteps. Conversely, the durations for LBF are somewhat shorter, with the active duration uniformly sampled between 15 and 25 timesteps, and the dead duration between 10 and 20 timesteps.

The teammate policies adhere to the experimental settings used for testing GPL (Rahman et al., 2021), which encompass a range of heuristic policies and pre-trained policies. Specifically, for Wolfpack, the teammate set includes the following agents: random agent, greedy agent, greedy probabilistic agent, teammate-aware agents, GNN-Based teammate-aware agents, graph DQN agents, greedy waiting agents, greedy probabilistic waiting agents, and greedy team-aware waiting agents. In the case of LBF, a combination of heuristics and A2C agents is employed as the teammate policy set. For more detailed information about teammate policies, we recommend referring to Appendix B.4 of GPL's paper.

In our investigation of different agent-type sets within LBF experiments (see Appendix I.2), we deliberately exclude the A2C agent from the original agent-type set, thereby establishing a distinct agent-type subset. It's crucial to acknowledge that the A2C agent provided by GPL is designed for scenarios with a maximum of 5 agents. Tailored to scenarios involving a greater number of agents, specifically up to 9, we undertake the additional step of training an A2C agent tailored to these expanded requirements.

In our experiments of studying the generalizability of CIAO, we constructed the agent-type sets for training and testing, respectively, for Wolfpack and LBF. The details are shown in Tab. 1.

Table 1: Variant agent-type sets for training and testing in experiments for evaluating generalizability of CIAO. The shorthand "Int" stands for the scenario where agent-type sets for training have intersection with testing. The shrothand "Exc" stands for the scenarios where agent-type sets for training are mutually exclusive to testing.

Scenario Name	Training	Testing
Wolfpack-Int	GreedyPredatorAgent, GreedyProbabilisticAgent, TeammateAwarePredator, DistilledCoopAStarAgent, GraphDQNAgent	GraphDQNAgent, RandomAgent, GreedyWaitingAgent, GreedyProbabilisticWaitingAgent, TeammateAwareWaitingAgent
Wolfpack-Exc	GreedyPredatorAgent, GreedyProbabilisticAgent, TeammateAwarePredator, DistilledCoopAStarAgent, GraphDQNAgent	RandomAgent, GreedyWaitingAgent, GreedyProbabilisticWaitingAgent, TeammateAwareWaitingAgent
LBF-Int	H8, H7, H6, H5, A2C0	A2C0, H1, H2, H3, H4
LBF-Exc	H8, H7, H6, H5, A2C0	H1, H2, H3, H4

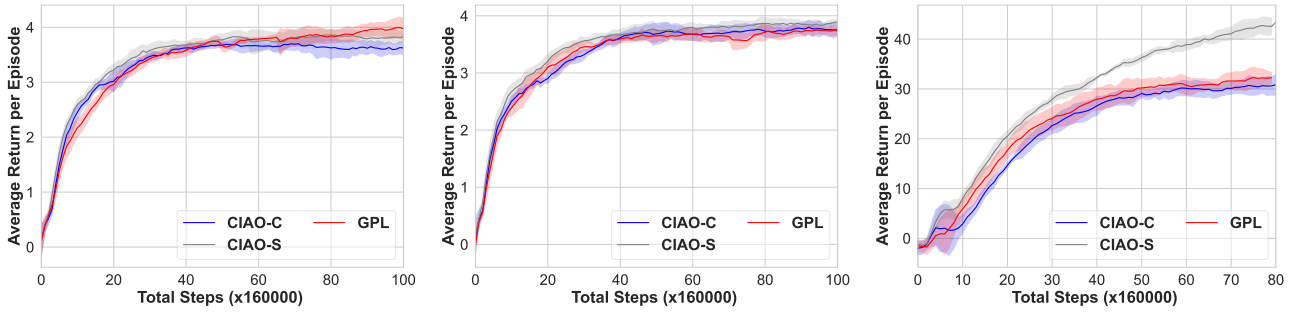
H.1. Detailed Hyperparameters and Computing Resources

We summarize the values of the common hyperparameters of algorithms that are used in our experiments, as shown in Tabs. 2 and 3. The optimizer we use during training is Adam (Kingma & Ba, 2014), with the default hyperparameters except learning rate. All algorithms in experiments are implemented in PyTorch (Paszke et al., 2019).

Then, we list the exclusive hyperparameters of all algorithms implemented in this work, as shown in Tab. 4. All experiments have been run on Xeon Gold 6230 with 30 CPU cores and 30 GB primary memory. An experiment conducted on Wolfpack requires approximately 11 hours, whereas an experiment on LBF typically takes around 12 hours.

I. Additional Experimental Results

I.1. Additional Evaluation on Small Number of Agents



(a) LBF including A2C agent. (b) LBF excluding A2C agent. (c) Wolfpack.

Figure 8: Comparison between CIAO and GPL in evaluation, across different scenarios of a maximum of 3 agents.

We present a performance comparison between CIAO and GPL across various scenarios involving a maximum of 3 agents, as illustrated in Fig. 8. The results indicate comparable performances on LBF, while CIAO-S significantly outperforms the other algorithms in the Wolfpack scenario. This observation leads to the conclusion that the star graph structure is better suited for Wolfpack. The rationale behind this outcome is that, in instances with a small number of agents in Wolfpack, conveying the learner's 'instructions' through one teammate to another is less effective. This contrasts with the scenario depicted in Fig. 3b,

Table 2: Shared hyperparameters for LBF. Note that the arguments `intersection_generalization`, `exclusion_generalization` and `exclude_A2Cagent` cannot be simultaneously set to be `True`.

Hyperparameter	Value
lr	0.00025
gamma	0.99
max_num_steps	1000000
eps_length	200
update_frequency	4
saving_frequency	50
pair_comp	bmm
num_envs	16
tau	0.001
eval_eps	5
weight_predict	1.0
num_players_train	3
num_players_test	5 for a maximum of 5 agents 9 for a maximum of 9 agents
exclude_A2Cagent	True for the agent-type set excluding A2C agent False for the default agent-type sets
intersection_generalization	True for the agent-type sets for training and testing are intersected False for the default agent-type sets
exclusion_generalization	True for the agent-type sets for training and testing are mutually exclusive False for the default agent-type sets
seed	0
eval_init_seed	2500

Table 3: Shared hyperparameters for Wolfpack. Note that the arguments `intersection_generalization` and `exclusion_generalization` cannot be simultaneously set to be `True`.

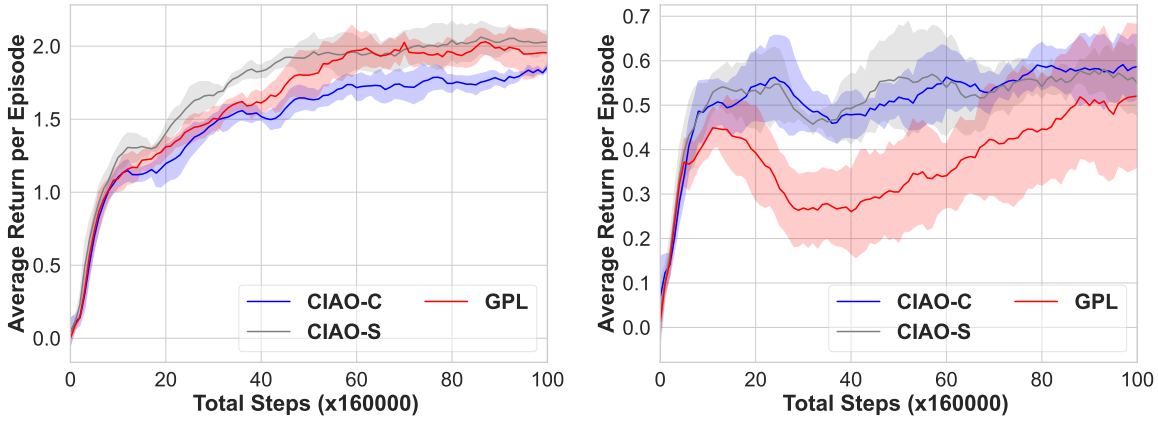
Hyperparameter	Value
lr	0.00025
gamma	0.99
num_episodes	4000
update_frequency	4
saving_frequency	50
pair_comp	bmm
num_envs	16
tau	0.001
eval_eps	5
weight_predict	1.0
num_players_train	3
num_players_test	5 for a maximum of 5 agents 9 for a maximum of 9 agents
intersection_generalization	True for the agent-type sets for training and testing are intersected False for the default agent-type sets
exclusion_generalization	True for the agent-type sets for training and testing are mutually exclusive False for the default agent-type sets
seed	0
eval_init_seed	2500
close_penalty	0.5

where a larger number of agents necessitates transmitting the learner's instructions through an intermediary teammate. The consistency of these findings reinforces the argument for the star graph structure's superiority in Wolfpack scenarios.

Table 4: Exclusive hyperparameters of all algorithms implemented in this paper.

Algorithm	weight_regularizer	graph	pair_range	indiv_range
GPL	0.0	complete	free	free
CIAO-S	0.5	star	pos	pos
CIAO-S-NP	0.5	star	neg	pos
CIAO-S-FI	0.5	star	pos	free
CIAO-S-ZI	0.5	star	pos	zero
CIAO-S-NI	0.5	star	pos	neg
CIAO-C	0.5	complete	pos	pos
CIAO-C-NP	0.5	complete	neg	pos
CIAO-C-FI	0.5	complete	pos	free
CIAO-C-ZI	0.5	complete	pos	zero
CIAO-C-NI	0.5	complete	pos	neg

I.2. LBF with Agent-Type Sets Excluding A2C Agent



(a) LBF: max. of 5 agents.

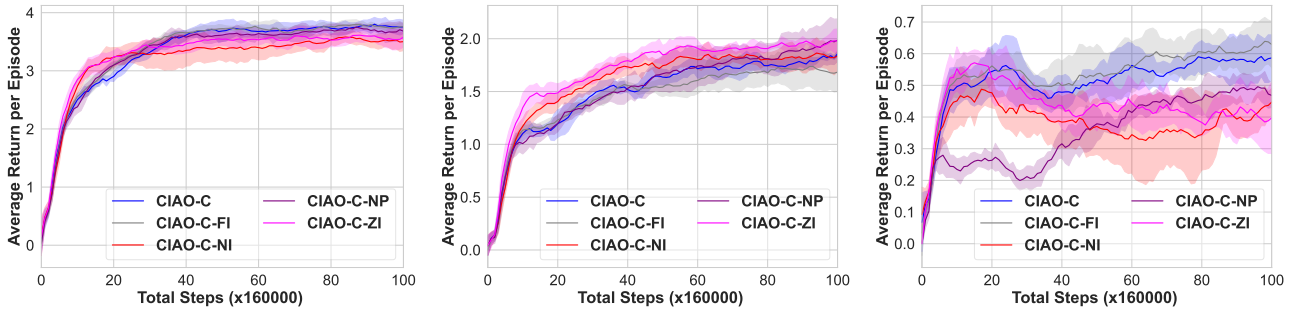
(b) LBF: max. of 9 agents.

Figure 9: Comparison between CIAO and GPL on LBF with the agent-type set excluding the agent-type generated by RL algorithms, in scenarios with a maximum of 5 and 9 agents.

We extend our evaluation of CIAO to LBF considering the agent-type set without the agent-type trained by RL (A2C agent), as depicted in Fig. 9. A comparison between Fig. 9 and Fig. 3 leads to the conclusion that CIAO-S exhibits comparatively robust performance across different agent-type sets, whereas CIAO-C demonstrates robustness primarily in scenarios with a larger number of agents. The underlying reasons for CIAO-C's limited robustness in situations with a small number of agents remain a topic for future investigation. Additionally, exploring the correlation between the performance of these algorithms in testing and RL-based agent-types is a valuable topic for further research.

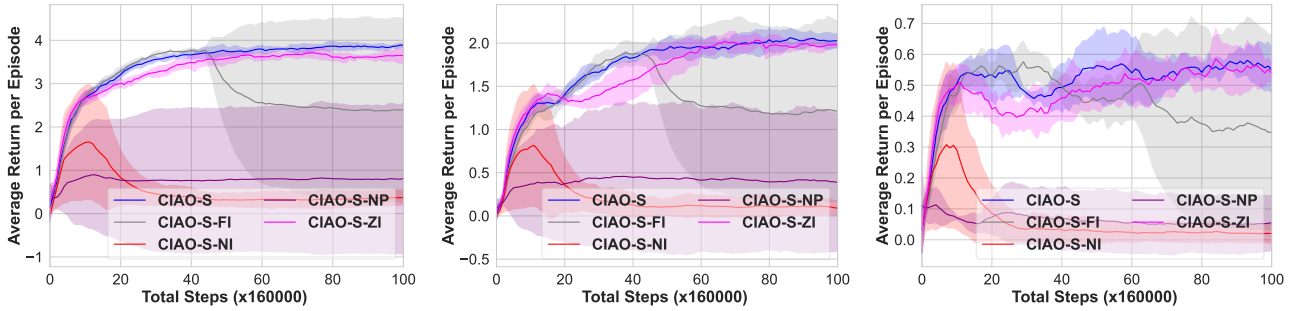
I.3. Additional Ablation Study on LBF with Agent-Type Sets Excluding A2C Agent

We present a comprehensive performance comparison among CIAO-C, CIAO-S, and their respective ablation variants on LBF, excluding the A2C agent. Figs. 10 and 11 illustrate the results for CIAO-C and CIAO-S, respectively. In the majority of situations, our hypothesis regarding the non-negative individual utility range is validated. However, we note that the unregularized individual utility exhibits satisfactory performance but is prone to instability. Additionally, our theoretical expectation of a non-negative pairwise utility range is violated for CIAO-C in scenarios involving a maximum of 3 and 5 agents. The root cause of this deviation requires further investigation, suggesting a potential avenue for future research into dynamic affinity graph structures.



(a) Maximum of 3 agents. (b) Maximum of 5 agents. (c) Maximum of 9 agents.

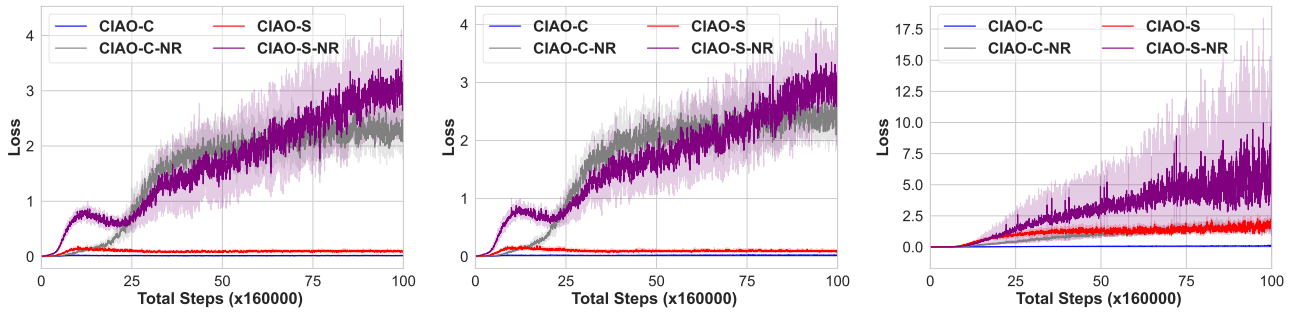
Figure 10: Comparison between CIAO-C and its ablations in evaluation, on LBF excluding A2C agent, across scenarios of various maximum numbers of agents as 3, 5 and 9, respectively.



(a) Maximum of 3 agents. (b) Maximum of 5 agents. (c) Maximum of 9 agents.

Figure 11: Comparison between CIAO-S and its ablations in evaluation, on LBF excluding A2C agent, across scenarios of various maximum numbers of agents.

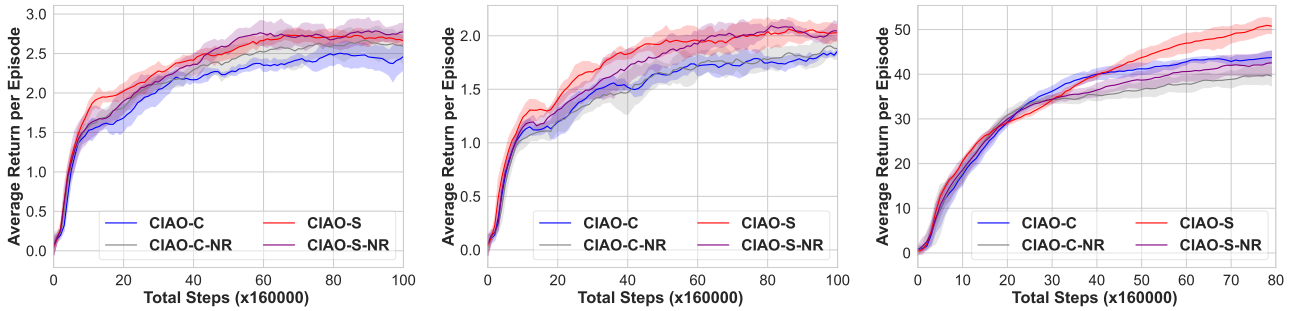
I.4. Additional Ablation Study on CIAO with No Regularizers



(a) LBF including A2C agent. (b) LBF excluding A2C agent. (c) Wolfpack.

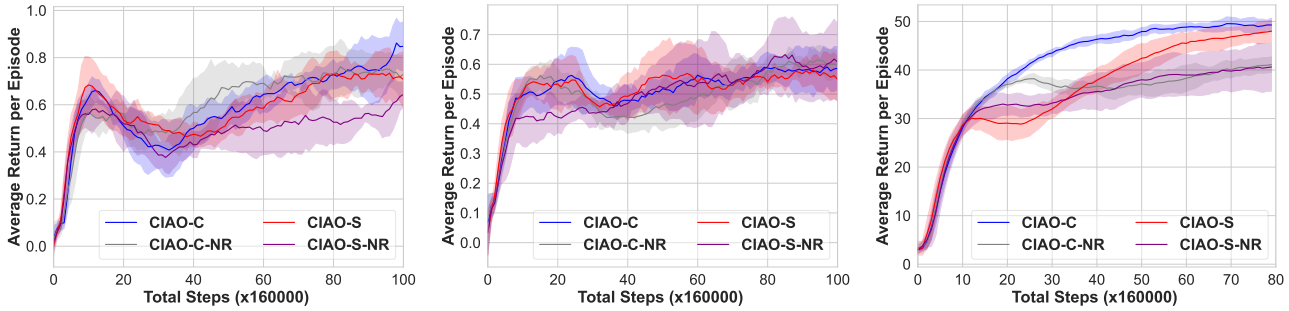
Figure 12: Comparison between CIAO and its ablation variant with no consideration of regularizers, denoted as CIAO-X-NR ("X" is either "C" or "S"), on the regularization loss during training, across different scenarios where the training is conducted with a maximum of 3 agents.

We conduct a performance comparison between CIAO and its ablation variant, excluding considerations of regularizers. In Fig. 12, the regularization losses during training are depicted, affirming the importance of incorporating regularizers. Notably, the effectiveness of regularizers is not consistently robust in the context of LBF, as shown in Figs. 13 and 14. Two potential explanations arise: (1) unique properties of the LBF environment may diminish the impact of regularizers, and (2) the regularization, driven by a sufficient condition to address DVSC as an RL problem, may lack consideration of other



(a) LBF including A2C agent. (b) LBF excluding A2C agent. (c) Wolfpack.

Figure 13: Comparison between CIAO and its ablation variant with no consideration of regularizers, denoted as CIAO-X-NR ("X" is either "C" or "S"), across different scenarios where the evaluation is conducted with a maximum of 5 agents.



(a) LBF including A2C agent. (b) LBF excluding A2C agent. (c) Wolfpack.

Figure 14: Comparison between CIAO and its ablation variant with no consideration of regularizers, denoted as CIAO-X-NR ("X" is either "C" or "S"), across different scenarios where the evaluation is conducted with a maximum of 9 agents.

eligible conditions. The exploration of these possibilities is deferred to the future research.