Learning to Collaborate with Unknown Agents in the Absence of Reward

Zuyuan Zhang[1], **Hanhan Zhou**[1], **Mahdi Imani**[2], **Taeyoung Lee**[1], **Tian Lan**[1]

[1]The George Washington University
[2]Northeastern University
zuyuan.zhang@gwu.edu, hanhan@gwu.edu, m.imani@northeastern.edu, tylee@gwu.edu, tlan@gwu.edu

Abstract

With the advancements of artificial intelligence (AI), emerging scenarios involving close collaboration between AI and other unknown agents are becoming increasingly common. This requires sometimes training AI agents to collaborate with unknown agents in the absence of a reward function – which may be unavailable to the AI agents or even undefined by the unknown agents themselves – thus posing news challenges to existing learning algorithms that often require knowing the shared reward. In this paper, we show that effective teaming with unknown agents can be achieved in the absence of a reward function, through actively modeling other unknown agents and reasoning about their latent rewards from available interaction/observation history. In particular, we propose a novel framework that leverages a kernel density Bayesian inverse learning method for active reward/goal inference and prove that multi-agent reinforcement learning guided by the inferred reward signals can converge to an optimal policy teaming with unknown agents. The result enables us to develop an adaptive policy update strategy, through the use of a family of pre-trained, goal-conditioned policies, further eliminating the need for online retraining. The proposed solution is evaluated using a wide range of diverse unknown agents of latent and even non-stationary reward. Our solution significantly increases the teaming performance between AI and unknown agents in the absence of reward.

Introduction

Advancements in artificial intelligence and machine learning (AI/ML) are enabling more and more scenarios where AI agents need to collaborate with other autonomous systems or humans, which may be unfamiliar entities (to the AI agents) with unknown goals/preferences (Johnson, Vignatti, and Duran 2020; Dafoe et al. 2020), denoted as unknown agents. Examples include teaming with autonomous agents that were built by other developers with unknown designs/parameters (Traeger et al. 2020; Albrecht and Stone 2018), or humans in a shared work environment with unknown intents/goals (Simmler and Frischknecht 2021; Behymer and Flach 2016). In the absence of a reward function – which may be unavailable to the AI agents (in the first example) or even undefined by the unknown agents themselves (in the second example) – training AI agents to collaborate with such unknown agents poses significant new challenges. Existing approaches like multi-agent reinforcement learning (MARL) (Spaan 2012) and transfer learning (Weiss, Khoshgoftaar, and Wang 2016) require knowing the shared reward function. Similarly, and hoc teamwork (Mirsky et al. 2022; Genter, Agmon, and Stone 2011; Papoudakis, Christianos, and Albrecht 2021) and zero-shot coordination (Treutlein et al. 2021; Li et al. 2023) cannot support collaboration with unknown agents if the reward is not known to the AI agents.

In this paper, we consider the problem of building AI agents for **S**ynergistic **T**eaming with **UN**known agents (STUN), in the absence of reward. This requires developing novel algorithms without relying on the common assumption of a known shared reward to guide learning. Specifically, our goal is to train a team of AI agents to collaborate with a given team of unknown agents, who jointly pursue some collaborative reward that is not known to the AI agents. We propose a novel framework that leverages active goal/reward inference and enables adaptive policy update following the inferred rewards. More precisely, we develop a Kernel Density Bayesian Inverse Learning (KD-BIL) (Mandyam et al. 2022) to obtain a posterior estimate of the latent reward of the unknown agents, from available observations/interactions history on the fly. The method is sample efficient and eliminates the need to refit policies for each sampled reward function by utilizing the kernel density estimation to approximate the likelihood function. Showing that KD-BIL produces unbiased estimates of the latent reward, we prove that MARL guided by the inferred rewards can converge to an optimal policy as if the true rewards of the unknown agents were available.

Interesting, our analysis show that using the maximum a posteriori (MAP) of latent rewards cannot ensure the optimality of learning collaborative policies in this STUN problem. We prove that obtaining an unbiased estimation is necessary to ensure convergence and optimality of the Bellman equation. Based on this result, we utilize the inferred rewards from KD-BIL to support a real-time policy adaptation, by pre-training a family goal-conditioned collaborative policies with respect to randomly sampled surrogate models. The family of goal-conditioned policies can be viewed as a parametric policy class. Thus, real-time policy adaptation can be achieved without any online re-training or learning by directly using the inferred reward. More precisely,

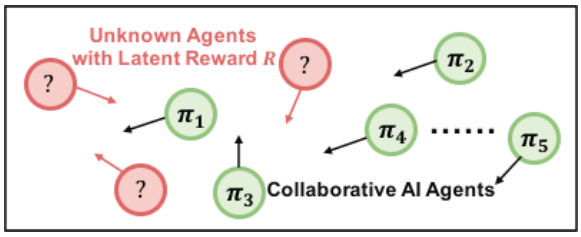


Figure 1: Our goal is to train policies π_i of collaborative AI agents to support a given team of unknown agents, when their reward R is not available (to the AI agents). It requires reasoning about unknown agents and their reward R from a Bayesian perspective and using the inferred results for learning optimal policies π_i .

our solution leverages a decentralized partially observable Markov Decision Process (dec-POMDP) model (Oliehoek, Amato et al. 2016) to pre-train collaborative agent policies with respect to randomly sampled surrogate models (of the unknown agents), such that the learned collaborative policies $\pi_i(a_i|s_i, R)$ (known as the goal-conditioned policies) are conditioned on potential rewards R. To team up with unknown agents, an adaptive policy update can be easily achieved by conditioning the learned collaborative policies on the unbiased posterior estimate $\hat{R}$, i.e., $\pi_i(a_i|s_i, \hat{R})$. The proposed solution supports centralized pre-training and fully decentralized executions (Kraemer and Banerjee 2016).

To validate the effectiveness of our framework, we create new environments for teaming and collaborating with unknown agents by modifying the MPE and SMAC environments (Lowe et al. 2017; Whiteson et al. 2019). We redesigned the reward system to reflect various latent play styles and unknown rewards of the unknown agents. Collaborative AI agents are then deployed in these shared environments together with given unknown agents (e.g., downloaded from public repositories or trained with latent rewards using popular MARL algorithms like MAPPO, IPPO, COMA, and IA2C (Yu et al. 2022; Schulman et al. 2017; Foerster et al. 2018; Mnih et al. 2016)). We note that our selected baselines in evaluation primarily include MARL algorithms and multi-task learning algorithms without relying on knowing the reward during execution. We do not consider other algorithms that fail to operate in the absence of reward (Weiss, Khoshgoftaar, and Wang 2016). We show that STUN agents consistently achieve close-to-optimal teaming performance with unknown agents. On super hard maps like $27m\_vs\_30m$ or $mmm2$, our solution improves the reward of unknown agents by up to 50%. The effectiveness of our design demonstrated through ablation studies. The evaluations also demonstrate our algorithm's ability to work with non-stationary unknown agents.

Related Work

Human AI teaming: Existing work on human-AI teaming often focus on teaming dynamics and organizational behavior contributes to understanding how to build effective human-AI teams (Dafoe et al. 2020; Albrecht and Stone 2018), leveraged cognitive science to better model and complement human decision-making processes (Hu and Sadigh 2023; Traeger et al. 2020), and considered related issues such as communication, trust, and collaboration strategies (Bauer et al. 2023). Recent work also includes zero-shot RL methods for policy adaptation and teaming, e.g., zero-shot human-agent coordination (Zhao et al. 2023) using a maximum entropy-based population training method, generating diverse cooperative agents to enhance AI adaptability in teams (Charakorn, Manoonpong, and Dilokthanakul 2022), and learning zero-shot cooperation with humans under the assumption of human biases (Yu et al. 2023). In contrast, we consider human as unknown agents and learn to collaborate with them in the absence of reward.

Multi-agent Reinforcement Learning: Most of the successful RL applications, e.g., gaming (Iqbal and Sha 2019; Foerster et al. 2017) and robotics (Knapp, Hall, and Horgan 2013; Robinette et al. 2016), involve the participation of multiple agents. For collaborative agents (Matignon, Laurent, and Le Fort-Piat 2007; Panait, Sullivan, and Luke 2006), MARL learns joint decision-making policies to optimize a known shared reward. These problems are often solved using policy- or value-based methods like MAPPO (Yu et al. 2022), MATRPO (Li and He 2023), and factorization (Rashid et al. 2020; Zhou, Lan, and Aggarwal 2022). Existing work also include transfer learning in this context (Yang et al. 2021). However, MARL algorithms require knowing the shared reward and cannot work when teaming with unknown agents in the absence of reward .

Multi-task Learning: Multi-task learning aims to train one policy that has the ability to solve multiple tasks simultaneously (Caruana 1997). This is often achieved by training a generalized policy against tasks randomly sampled from some task space. Various methods have been proposed to support multi-task RL, e.g., attention mechanisms (Niu, Zhong, and Yu 2021; Vezhnevets et al. 2017; Wang et al. 2023) and transferring policies between multi-tasks (Rusu et al. 2016; Omidshafiei et al. 2017; Gurulingan, Zonooz, and Arani 2023). While multi-task Learning can produce generalized policies that work reasonably well in different task environments, e.g., alongside different unknown agents without knowing their rewards during execution, they cannot ensure optimal teaming performance toward one particular unknown agent.

Inverse Reinforcement Learning: Inverse Reinforcement Learning (IRL) infers goals/rewards from observations of action trajectories. It was first proposed in (Ng, Russell et al. 2000) showing that IRL problem has infinite solutions. Several solutions, such as MaxEntropy IRL (Ziebart et al. 2008), Max-Margin IRL (Abbeel and Ng 2004), and Bayesian IRL (Ramachandran and Amir 2007) have been proposed to resolve the ambiguity of IRL. In contrast, kernel-based IRL in this paper is more sample efficient and supports synergistic teaming with unknown agents in the absence of reward.

Our Proposed Solution

Problem Statement

We consider a set $\mathcal{N}$ of n collaborative AI agents (also denoted as STUN agents in this paper), which need to sup-

port a given set $\mathcal{N}_u$ of n_u unknown agents toward some latent goal, i.e., the reward of the unknown agents is not available to the AI agents. The decision making is modeled as a dec-POMDP (Oliehoek, Amato et al. 2016), given by a tuple $\mathcal{M} = \langle \mathcal{N}_u, \mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{T}, R, \Omega, \mathcal{O}, \gamma \rangle$. $\mathcal{S}$ is the joint state space for all agents $\mathcal{N}_u \cup \mathcal{N}$. For each agent i, A_i is its action space and O_i its observation space. Thus, $\mathcal{A} = A_1 \times A_2 \times \cdots \times A_{n_u+n}$ denotes the joint action space of all agents, and $\mathcal{O} = O_1 \times O_2 \times \cdots \times O_{n_u+n}$ denotes the joint observation space of all agents. We use $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Omega(\mathcal{S})$ to denote the transition probability function, with $\Omega(\mathcal{S})$ representing a set of state distributions. We note that our formulation is different from existing work on ad hoc teamwork (Mirsky et al. 2022; Genter, Agmon, and Stone 2011) and zero-shot coordination (Treutlein et al. 2021; Li et al. 2023), since the reward is not available to the AI agents.

Each unknown agent $i \in \mathcal{N}_u$ behaves according to some latent policy $\pi_i^u : O_i \times A_i \to [0, 1]$, which maximizes some latent reward R not known to the AI agents. More precisely, the latent reward function $R : \mathcal{S} \times A \to \mathbb{R}$ is unknown and non-observable to other AI agents operating in the shared task environment, while their behavior trajectories are observable for reasoning and inferring the latent reward (which could be time-varying).

Our goal in this paper is to learn the policies of collaborative AI agents (i.e., STUN agents): $\pi_i : O_i \times A_i \to [0, 1]$ for $i \in \mathcal{N}$ in the absence of reward, in order to effectively team up with the unknown agents and collaboratively maximize the expected return $G = \sum_t^H \gamma^t R^t$, where γ is a discount factor, R^t is the latent reward received at time t, and H is the time horizon. It is easy to see that while the problem follows a dec-POMDP structure, it cannot be solved with existing solutions because training the AI agents would normally require having access to the latent reward signal R, and thus is not possible in our formulation with unknown agents. The collaborative AI agents must simultaneously address two problems: (i) inferring the latent reward by collecting observations of the unknown agents in the shared task environment, and (ii) adapting their policies on the fly to support effective teamwork toward the learned reward, while ensuring optimality of the learned MARL policies.

Learning with Active Goal Inference

We consider a team of STUN agents that are deployed alongside unknown agents in a shared task environment. The STUN agents can observe the trajectories $\{\tau^u\} = \{(o_i^u, a_i^u)\}_{i=1}^n$ of the unknown agents (defined as a sequence of their joint observations $o_i^u \in \mathcal{O}^u = O_1 \times O_2 \times \cdots \times O_{n_u}$ and joint actions $a_i^u \in \mathcal{A}^u = A_1 \times A_2 \times \cdots \times A_{n_u}$). We propose KD-BIL for active goal inference and show that MARL guided by the inferred reward signals can learn an optimal MARL policy. Further, to support quick adaptation without online retraining, we leverage a family of pre-trained, goal-conditioned for adaptive policy update.

Our proposed KD-BIL is a method for approximating the reward probability distribution using kernel density estimates. It not only allows an efficient estimate of the unknown agents' reward (which is often associated with substantial uncertainty) but also supports sample-efficient computations using limited observation data. Specifically, we set up a training dataset $\mathbb{D}$ by sampling m demonstrations, either from available trajectories of known agents with observable reward or by training surrogate unknown agents using sampled reward functions. This dataset consists of m 3-tuple demonstrations $\{(o_j, a_j, R_j)\}_{j=1}^m$, where R_j is the reward function used to generate the demonstrations (o_j, a_j). Using either known agents or surrogate agent models, we construct the training dataset that contains demonstrations of various potential behavior.

Given observed unknown agent trajectories (o_i^u, a_i^u) of size n, we can now estimate the posterior $p_m(R|o_i^u, a_i^u)$ by formulating conditional density $\hat{p}_m(o_i^u, a_i^u|R)$ with respect to the training dataset and our choice of kernel density function $K(\cdot, \cdot)$. Using demonstrations in the training dataset, the conditional density for a state-action pair (o_i^u, a_i^u) given a latent reward function R is

$$\hat{p}_m(o_i^u, a_i^u|R) \propto \sum_{j=1}^{m} \frac{K((o_i^u, a_i^u), (o_j, a_j)) \cdot K_R(R, R_j)}{\sum_{l=1}^m K_R(R, R_l)} \tag{1}$$

where $K(\cdot, \cdot)$ and $K_R(\cdot, \cdot)$ are two different kernel density functions for the state-action pair and for the reward, respectively. The proposed KD-BIL method works with any kernel functions, such as the Gaussian kernel and the Matern kernel (Mandyam et al. 2022). We consider the Gaussian kernel function in the following derivations. This implies that the conditional density $\hat{p}_m(o_i^u, a_i^u|R)$ is given by:

$$\begin{aligned} &\hat{p}_m(o_i^u, a_i^u|R) \\ &\propto \sum_{j=1}^{m} \frac{e^{-d_s((o_i^u, a_i^u), (o_j, a_j))^2/(2h)} e^{-d_r(R, R_j)^2/(2h')}}{\sum_{l=1}^m e^{-d_r(R, R_l)^2/(2h')}} \end{aligned} \tag{2}$$

where $d_s : (\mathcal{O} \times \mathcal{A}) \times (\mathcal{O} \times \mathcal{A}) \to \mathbb{R}$ is a distance metric to compare (o, a) tuples, $d_r : R \times R \to \mathbb{R}$ is a distance metric to compare reward functions, and $h, h' > 0$ are smoothing hyperparameters. We note that these hyperparameters can be further optimized using the training dataset, e.g., similar to optimizing surrogate models in Bayesian Optimization (Ramachandran and Amir 2007). Next, we obtain the following posterior estimate of latent reward.

Lemma 1. *The estimated posterior of unknown agent reward is given by*

$$\begin{aligned} &\hat{p}_m(R|\{o_i^u, a_i^u\}_{i=1}^n) \\ &\propto p(R) \prod_{i=1}^{n} \sum_{j=1}^{m} \frac{e^{-\frac{d_s((o_i^u, a_i^u), (o_j, a_j))^2}{(2h)}} e^{-\frac{d_r(R, R_j)^2}{(2h')}}}{\sum_{l=1}^m e^{-d_r(R, R_l)^2/(2h')}} \end{aligned} \tag{3}$$

Importantly, we note that this conditional density $\hat{p}_m$ can be easily computed from the training dataset and using kernel density functions. There is no need to refit policies or perform value iterations to evaluate the posterior for a given reward function R. This drastically reduces the computational complexity compared to existing Bayesian IRL algorithms (Zintgraf et al. 2021; Mandyam et al. 2022; Ramachandran and Amir 2007; Choi and Kim 2012; Chan and

van der Schaar 2021) and thus makes it possible to perform Bayesian inverse learning for the unknown agent's latent reward, even in complex environments with large state spaces (as shown in our evaluation on SMAC (Whiteson et al. 2019)). Further, $p(R)$ in Equation (3) denotes the prior distribution of reward functions. It can be a uniform distribution or estimated from available unknown agent statistics. As $m \to \infty$, it is shown that $\hat{p}_m$ converges to the true likelihood of reinforcement learning policies and thus the true posterior using results in (Van der Vaart 2000).

Lemma 2. *Assume the prior for R, denoted by $\mathcal{P}$, satisfies $\mathcal{P}(\{R : KL(R^*, R) < \epsilon\}) > 0$ for any $\epsilon > 0$, where KL is the Kullback–Leibler divergence. According to (Mandyam et al. 2023) As $m \to \infty$, the posterior measure corresponding to the posterior density function $\hat{p}_m$(3). denoted by $\mathcal{P}_m^n$, is consistent w.r.t. the L1 distance; that is,*

$$\lim_{\substack{m\to\infty \\ n\to\infty}} \mathcal{P}_m^n(\{\theta : ||\hat{p}(\cdot|R) - p(\cdot|R^*)||_{L_1} < \epsilon\}) = 1 \qquad (4)$$

Guided by the inferred reward using KD-BIL, we can utilize MARL to train a collaborative policy for teaming with unknown agents. We show that since the above Lemma guarantees that our KD-BIL algorithm is guaranteed to converge to the true reward distribution, the learned policy is guaranteed to be optimal. We prove the result using the Bellman equation with inferred rewards. In practice, we can consider a general representation of the latent reward function, i.e., $R_{\mathcal{B}}(s, a)$ with latent parameters $\mathcal{B} \in \mathbb{R}^k$, which k is the latent dimension of $\mathcal{B}$. This representation captures latent reward that can be expressed as a linear function $R_{\mathcal{B}}(s, a) = \mathcal{B}^T\mathbf{R}(s, a)$ of possible underlying components $\mathbf{R} = (R_1, R_2, \ldots, R_k)$ of the unknown agents' potential objectives, as well as more general forms of reward functions that are defined through a neural network $R_{\mathcal{B}}(s, a)$ parameterized by $\mathcal{B}$. For instance, in our evaluations using the MPE environment (Lowe et al. 2017), we can construct a mixing network (e,g., a single-layer neural network parameterized by $\mathcal{B}$) to combine underlying components such as greedy, safety, cost, and preference, into a more complex and realistic reward function representation for goal inference. Thus, active goal inference in this paper aims to estimate the latent reward parameters $\mathcal{B}$ from observed unknown agents' trajectories $\{(o_i^u, a_i^u)\}_{i=1}^n$, using the proposed KD-BIL method, by estimating posterior $\hat{p}_m(R_{\mathcal{B}}|\{o_i^u, a_i^u\}_{i=1}^n)$ over the latent reward parameters $\mathcal{B}$ instead.

To establish optimality of the proposed approach, we consider a Q-learning algorithm (Watkins and Dayan 1992) that are often employed for theoretical analysis. We consider learning over the joint action and state space (of the AI agents) and under the unbiased reward estimates $R_{\tilde{\mathcal{B}}}$. Thus, the Q-values, $Q_{\mathcal{B}}^{\pi}(s, a) = \mathbb{E}_{\pi}[\sum_t^H \gamma^t R^t | S_t = s, A_t = a, R^t \sim R_{\mathcal{B}}]$, are now defined with respect to the reward estimates $R_{\tilde{\mathcal{B}}}$. We show that this joint Q-learning guided by the estimate rewards converges to optimum.

Theorem 1. *(Inferred Reward using KD-BIL Ensures Optimality.) Given a finite MDP, denoting as $\mathcal{M}$, the Q-learning algorithm with inferred reward from KD-BIL, given by the update rule,*

$$Q_{\tilde{\mathcal{B}},t+1}(s_t, a_t) = (1 - \alpha_t)Q_{\tilde{\mathcal{B}},t}(s_t, a_t) + \alpha_t[R_{\tilde{\mathcal{B}}} + \gamma \max_{b\in\mathcal{A}} Q_{\mathcal{B},t}(s_{t+1}, b)] \qquad (5)$$

converges w.p.1 to the optimal Q-function as long as $\sum_t \alpha_t = \infty$ and $\sum_t \alpha_t^2 < \infty$

Note that a and s are the joint action and state of all AI agents. The right hand of Equation (5) relies on the inferred unbiased reward $R_{\tilde{\mathcal{B}}}(s, a)$. Theorem 1 states that collaborative agent policies will converge to optimum $w.p.1$ using the inferred rewards. According to Lemma 2, we can show that the reward is unbiased, i.e., $\mathbb{E}[R_{\tilde{\mathcal{B}}}] = R_{\mathcal{B}}$. Consequently, we can eliminate the bias introduced by the reward during each update of the Q-learning process, thereby achieving convergence. The inferred rewards and the resulting collaborative policies are evaluated in MPE and SMAC environments, validating the proposed methods.

Adaptive Policy Update with Inverse learning

While training MARL policies guided by the inferred reward signals ensures convergence to optimum, online re-training and re-learning may incur overhead, making the solutions not suitable for scenarios requiring real-time interaction and adaptions. To this end, we further propose an adaptive policy update to avoid such overhead. The key idea is to pre-train a class of goal-conditioned policies $\pi(a|o, R)$ for the STUN agents, which are conditioned on potential reward functions R. This pre-training is supported by the use of surrogate unknown agent models with respect to randomly sampled latent reward function R, as shown in Figure .2. The pre-trained STUN agents (with goal-conditioned policies $\{\pi_i(a_i|o_i, R)\}$) are then deployed in a shared task environment to collaborate with unknown agents and to support common goals/objectives. Using the inferred reward signals $\hat{R}$ from KD-BIL, we can easily adapt the pre-trained agent policies into $\{\pi_i(a_i|o_i, \hat{R})\}$, which ensures optimal teaming performance. It eliminates the need to refit or re-retraining MARL policies on the fly after active goal inference.

We show that a maximum a posteriori (MAP) of latent reward cannot ensure convergence of the Bellman equation to achieve optimal expected return $G = \sum_t^H \gamma^t R^t$. Instead, an unbiased estimate $\tilde{R}$ is proven to be necessary for learning the optimal collaborative policy. Based on this result, we leverage the pre-trained, goal-conditioned policies $\{\pi_i\}$ and perform adaptive policy update using the unbiased estimate, i.e, $\{\pi_i(a_i|o_i, \tilde{R})\}$. The proposed policy adaptation requires no re-training or re-learning, while ensuring optimality of the adapted policy. Again, we consider a general representation of the latent reward function, i.e., $R_{\mathcal{B}}(s, a)$ with latent parameters $\mathcal{B} \in \mathbb{R}^k$, which k is the latent dimension of $\mathcal{B}$.

We pre-train a set of **goal-conditioned policies** $\{\pi_i(a_i|o_i, \mathcal{B})\}$ – which are now conditioned on the latent reward function parameters $\mathcal{B}$ instead – for the collaborative AI agents. The pre-training is illustrated in Figure 2, where surrogate models of the unknown agent are leveraged. We employ MARL to train the surrogate models and the collaborative AI agent policies, using

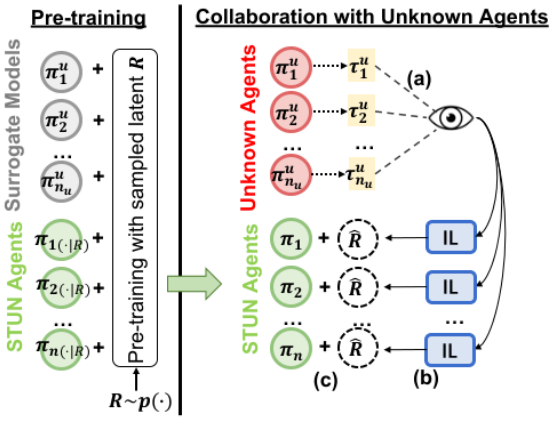


Figure 2: An illustration of our proposed framework. STUN agents $\pi_i(\cdot|R)$ are pre-trained using surrogate agent models with sampled latent rewards R. To collaborate with unknown agents, they use inverse learning (step b) on the observed trajectories $\{\tau_i^u\}$ of the unknown agents (step a) and perform adaptive policy update based on an unbiased estimate $\hat{R}$ (step c).

reward signals $R_{\mathcal{B}}$ with randomly sampled parameters $\mathcal{B} \sim p(\cdot)$. Thus, an adaptive policy update can be achieved during teaming stage by feeding the unbiased estimate $\tilde{\mathcal{B}}$ (of unknown agent reward) into the goal-conditioned policies, i.e., $\{\pi_i(a_i|o_i, \tilde{\mathcal{B}})\}$. It ensures optimal teaming performance with the unknown agents.

The pseudo-code of our proposed STUN framework can be found in Appendix A. In particular, the pre-training of goal-conditioned policies (together with surrogate models) can leverage any MARL algorithms to maximize the expected return $J_{\mathcal{B}}(\theta)$ under sampled reward:

$$J_{\pi,\mathcal{B}}(\theta) = \mathbb{E}_{s_0\sim\mu,s,a}[V_{\mathcal{B}}^{\pi_\theta}(s_0)] \tag{6}$$

where s_0 is drawn from the initial state distribution. In this paper, we use policy gradient algorithms to pre-train the goal-condition policies. So, the Latent Style Policy Gradient for pre-training can use this way $\nabla_\theta J_{\pi,\mathcal{B}}(\theta) = \mathbb{E}_{\pi_\theta}[Q_{\mathcal{B}}^{\pi_\theta}(s,a)\nabla_\theta \log \pi_\theta(a|o,B)]$

Experiments

We redesign two multi-agent simulation environments based on multiagent-particle-envs(MPE) (Lowe et al. 2017) and SMAC (Whiteson et al. 2019) to create collaborative teaming tasks with unknown agents. These environments include blue (friendly) and red (adversarial) teams. The red team is controlled by built-in policies within the environments, while the blue team consists of both unknown agents and collaborative AI agents. We consider different methods for creating the unknown agents, such as training with randomly-generated latent reward parameters and using available agents from popular MARL algorithms like MAPPO, IPPO, COMA, and IA2C (Yu et al. 2022; Schulman et al. 2017; Foerster et al. 2018; Mnih et al. 2016). We deploy STUN agents (and other baseline agents like obtained by multi-task learning) alongside these unknown agents and evaluate the teaming performance in each collaborative task. We note that the baselines selected in our evaluation primarily include MARL algorithms and multi-task learning algorithms without relying on knowing the reward during execution. We do not consider other algorithms that fail to operate in the absence of reward (Weiss, Khoshgoftaar, and Wang 2016).

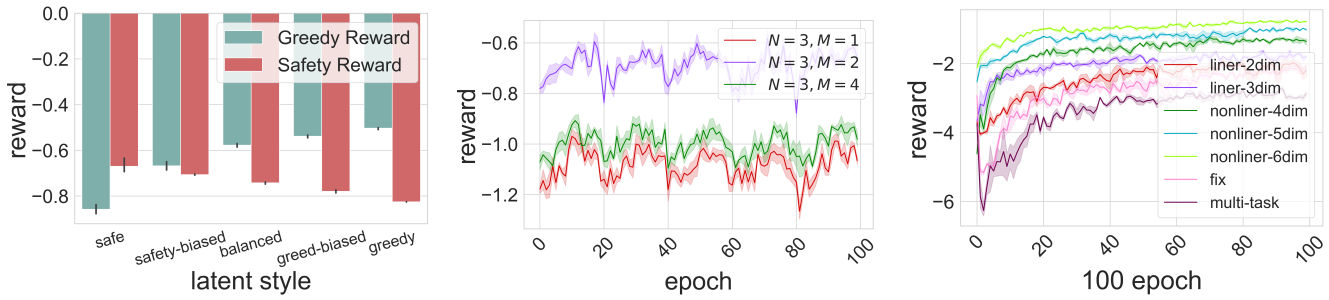
Multi-Agent Particle Environment

We first consider Predator-Prey from MPE environment, which is a partially observable multi-agent environment that involves N AI-controlled blue agents and M adversary agents. Half of the blue agents are unknown agents with latent reward, while the rest of blue agents are collaborative AI agents. Meanwhile, the adversaries follow a fixed strategy: they move towards the nearest agent, and different adversaries will choose different targets.

To create unknown agents with diverse behaviors/objectives, we consider four methods of creating reward components and combines them into more complex reward functions $R_{\mathcal{B}}(s,a)$ using either a linear function or a non-linear mixing network (e.g., a single-layer neural network) with latent parameters $\mathcal{B}$. We consider **(i) Greedy Reward:** As Preys get closer to each other, they are less greedy in terms of exploration, thus resulting in a negative reward. **(ii) Safety Reward:** Prey attempts to evade Predators and receives a negative reward proportional to the distance. **(iii) Cost Reward:** Movement by the Prey consumes energy, so when the Prey moves, it also receives a negative reward. **(iv) Preference Reward:** Different weights can be assigned to Predator-Pray pairs in the other classes, to reflect individual preferences/importance. We note that the combination of these methods allows creation of complex reward functions with many dimensions.

We evaluate the performance of STUN agents teaming up with unknown agents and focus on two key aspects: (1) Adaptability: evaluating whether trained STUN agents can maintain high teaming performance when collaborating with new, unknown agents or agents with time-varying behaviors/objectives. (2) Interpretability of behavior: Assessing how collaborative agents' behaviors vary under different unknown agents with latent $\mathcal{B}$. We also perform ablation studies to verify the impact of (i) adaptive policy update using goal-conditioned policies and (ii) active goal inference of our proposed design, as well as the impact of high-dimensional, linear, and nonlinear reward functions.

Teaming behavior interpretation. To better interpret STUN agent behaviors, we focus on Greedy Rewards and Safety Rewards in this experiment. While more details are provided in Appendix C.1.2, we show in Fig. 3(a)(a) the achieved tradeoff between the two reward components when collaborating with unknown agents of different latent $\mathcal{B}$. As the unknown agents tend to move from playing safe (i.e., staying away from predators) to being greedy (i.e., more aggressively exploring), STUN agents adapt their policies and also becomes more greedy – as shown by diminishing safety return and increasing greedy return. More teaming analysis



(a) Interpreting teaming behavior. (b) Against non-stationary unknown agents. (c) Ablation and scalability study.

Figure 3: (a) Illustration of the underlying reward tradeoff when STUN agents team up with unknown agents ranging from playing safe to being greedy. (b) Ability of STUN agents to quickly reasoning/inferring the time-varying reward of the unknown agents (changing every 20 epochs) and then performing adaptive policy update on the fly. (c) Ablation studies showing the impact of different design modules, as well as robust performance of STUN under unknown reward function with increasing complexity (e.g., increasing from 2 to 6 dimensions of reward components and using nonlinear mixing functions).

Unknown Agents	Synergistic Agents	Fixed-Behavior Agents			Multi-task Agents				
	STUN	FBA-C	FBA-B	FBA-A	MAPPO	IPPO	COMA	MAA2C	IA2C
FBA-C	**1.10**	1.04	1.04	0.99	0.88	0.85	0.84	0.69	0.68
FBA-B	**2.38**	2.06	2.36	2.19	1.99	1.84	1.83	1.71	1.73
FBA-A	3.73	2.88	3.55	**3.88**	3.08	2.94	2.68	2.35	2.67
Mixed-1	**1.98**	1.11	1.21	0.98	1.13	1.02	0.88	0.72	0.66
Mixed-2	**2.21**	1.43	1.36	1.87	1.73	1.66	1.11	1.23	1.02
Mixed-3	**2.96**	1.60	2.10	2.68	1.89	1.44	1.38	1.35	1.07
MAPPO	**2.43**	2.04	2.05	2.08	2.11	1.97	1.80	1.87	1.56
IPPO	**2.25**	1.82	2.22	2.19	1.95	2.22	2.06	1.84	1.87
COMA	2.12	1.62	1.86	1.95	1.98	1.81	**2.18**	1.55	1.58
MAA2C	**2.22**	1.58	1.97	2.05	1.25	1.83	1.79	2.13	2.12
IA2C	**2.08**	1.73	1.89	1.87	1.84	1.62	1.55	1.43	1.97
Average	**2.31**	1.72	1.96	2.07	1.80	1.75	1.65	1.53	1.54
Normalized	**98.9**	73.7	84.0	88.7	77.1	74.9	70.7	65.6	66.0

Table 1: Experimental results of average reward on `3s_vs_5z` map of the redesigned SMAC environment. STUN and selected baselines (during testing) are each teamed up with 11 different unknown agents, respectively. Each row represents the teaming performance with a different unknown agent. STUN achieves nearly optimal performance in nearly all scenarios.

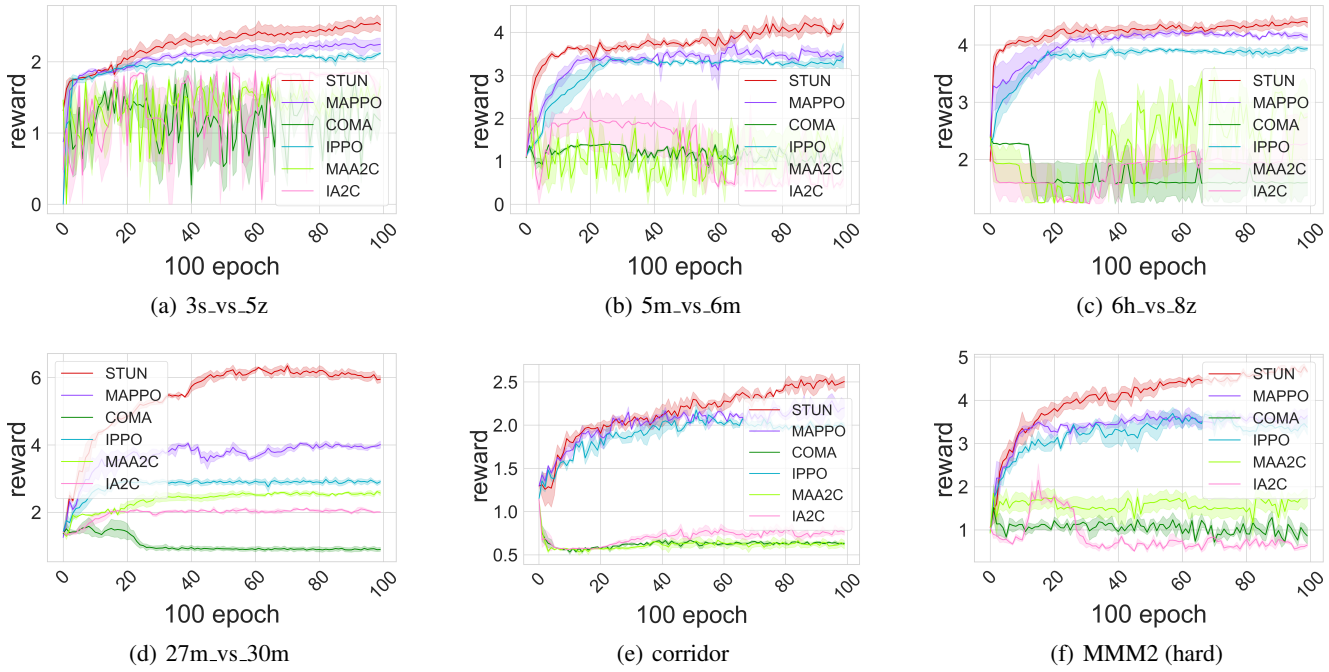
and illustrations are provided in Appendix C.1.2.

Collaborating with changing unknown agents. During the teaming/execution stage, we deploy trained STUN agents alongside unknown agents with changing behavior. Specifically, the unknown agents vary their policies at the beginning of every 20 epochs. This requires STUN agents to continually reason/infer the time-varying reward of the unknown agents and then perform adaptive policy update on the fly. Fig. 3(a)(b) shows that STUN agents can swiftly adapt their policies in just 5-10 epochs (with goal inference and adaptive policy update) and ramp up teaming performance, in different environments with M adversaries.

Ablation studies. We now perform an ablation study to (i) remove the adaptive policy update in STUN agents by instead performing additional online reinforcement learning using the inferred reward and (ii) remove the active goal inference by conditioning STUN agents on fixed reward parameters – labeled "multi-task" and "fix" respectively in Fig. 3(a)(c). Significant performance degradations are observed comparing to STUN agents labeled "nonlinear-4dm". For scalability, in Fig. 3(a)(c), we further vary the dimensions of underlying reward components from 2 to 6 and evaluate STUN agents over both linear and non-linear reward functions (e.g., soft-max and single-layer network with parameters $\mathcal{B}$). The numerical results demonstrate STUN agents' robust teaming performance with increasingly complex unknown reward structures.

StarCraft Multi-Agent Challenge

In this section, we perform extensive evaluations of the proposed framework on SMAC tasks (e.g., hard and super-hard maps) and compare it with a range of baseline algorithms.



(a) 3s_vs_5z (b) 5m_vs_6m (c) 6h_vs_8z

(d) 27m_vs_30m (e) corridor (f) MMM2 (hard)

Figure 4: Performance comparison of our proposed STUN agents and selected baselines on redesigned SMAC tasks.

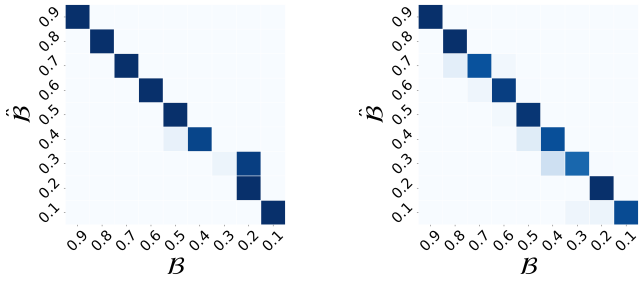
Note that to create unknown agents with different latent rewards, we have redesigned the SMAC environment[1] to consider two broad classes of rewards: **Conservative Rewards** that are represented by the health values of surviving friendly blue-team agents and **Aggressive Rewards** that are represented by the total damage inflicted on adversarial red-team agents. This design allows us to create diverse unknown agents with different latent reward function and play style, ranging from conservative to aggressive as parameterized by the latent $\mathcal{B}$. Teaming performance is measured using the achieved (latent) reward. All other environment settings remain the same as standard SMAC. Detailed information on our settings and training configurations like hyperparameters used can be found in the Appendix.

We consider 6 maps selected from SMAC with varying level of difficulty and create 7 types of unknown agents either by training with fixed unknown behaviors or by directly using agents from popular MARL algorithms including MAPPO, IPPO, COMA, and IA2C (Yu et al. 2022; Schulman et al. 2017; Foerster et al. 2018; Mnih et al. 2016). We deploy trained STUN agents in these collaborative tasks against each type of unknown agents and compare the performance to a number of baselines such as optimal fixed-behavior agents (e.g., with conservative (FBA-C), balanced (FBA-B), and aggressive (FBA-A) play styles) and collaborative agents that employ multi-task learning by randomly sampling the unknown agents' latent parameters. We also consider players with **mixed behaviors**: Mixed-1 consists of 50% FBA-C and 50% FBA-B. Mixed-2 consists of 50% FBA-C and 50% FBA-A. Mixed-3 consists of 50% FBA-B and 50% FBA-A. In the following evaluations, we will demonstrate: (1) The proposed KD-BIL can accurately infer latent reward parameters $\mathcal{B}$; (2) STUN agents can efficiently team up with unknown agents and outperform baselines on various SMAC tasks; and (3) STUN agents demonstrate robust performance with diverse unknown agent rewards.

Evaluating goal inference against ground truth. We validate the effectiveness of our proposed goal inference algorithm, KD-BIL, by showing the correlation between the estimate posterior and the ground truth (in terms of the latent reward parameter $\mathcal{B}$) in Fig.5 on two maps, $3s\_vs\_5z$ and *corridor*. Each row in the heatmap shows the posterior distribution of one given reward parameter (which in the ideal case would concentrate on the diagonal line). The result shows that our proposed goal inference can accurately estimate the latent reward, even in complex tasks that involve a large number of blue/red agents and require advanced strategies (e.g, on the *corridor* map). The analysis of goal inference on other maps are provided in the appendix.

Evaluating performance on different maps. We deploy trained STUN agents alongside unknown agents on 6 different maps and repeat the experiments with several SOTA baselines for comparison: MAPPO (Yu et al. 2022), COMA (Foerster et al. 2018), MAA2C, IPPO (Schulman et al. 2017), IA2C (Mnih et al. 2016). These baseline agents are trained using a multi-task learning approach by randomly sampling latent $\mathcal{B}$, so that they can collaborate with unknown agents of different behaviors/objectives. The results, as shown in Fig. 4, demonstrate that the STUN's pre-

[1]Standard SMAC environment considers only winning rate as the reward, which is insufficient for creating diverse unknown agents with latent rewards for our evaluation.



(a) 3s_vs_5z (b) MMM2

Figure 5: An illustration of the correlation between posterior estimate of reward parameters (shown in each row) using KD-BIL and the ground-truth reward parameters. Our proposed active goal inference can accurately infer the latent reward from observed unknown agent trajectories.

training method can effectively converge and significantly improve teaming performance (up to 50% on certain super-hard maps). The settings are detailed in the Appendix.

Teaming performance with various unknown agents. We deploy the trained STUN agents alongside 8 different unknown agents on $3s\_v\_5z$ map and compare the teaming performance against two groups of baselines – fixed-behavior agents and multi-task agents trained using different algorithms – which are also deployed alongside the same unknown agents. Table 1 summarizes numerical results, with each row comparing the teaming performance of various collaborative agents alongside the same unknown agent. In particular, we calculate a normalized teaming score by assigning 100 points to the best performing agent in each row and then taking the average over all 8 unknown agents. STUN agents achieve a normalized score of 99.2 out of a maximum of 100, with the best performance in nearly all scenarios and demonstrating robust teaming performance with a diverse range of unknown agents.

Conclusions

This paper presents a novel framework for enhancing AI and unknown-agent teaming in collaborative task environments in the absence of reward. Considering unknown agents whose rewards are not available to the AI agents, we show that the proposed KD-BIL provides unbiased reward estimates and thus ensures optimality of learned MARL policies under the guidance of inferred rewards. We then propose a goal-conditioned policy adaptation without re-training or learning. Evaluations using diverse unknown and also non-stationary agents in MPE and SMAC demonstrate robust teaming performance. Synergistic teaming with unknown agents in non-stationary tasks or under restricted observations are avenues for future work.

Acknowledgments

This work was partially supported by the Office of Naval Research under grant N00014-23-1-2850.

References

Abbeel, P.; and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, 1.

Albrecht, S. V.; and Stone, P. 2018. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258: 66–95.

Bauer, J.; Baumli, K.; Behbahani, F.; Bhoopchand, A.; Bradley-Schmieg, N.; Chang, M.; Clay, N.; Collister, A.; Dasagi, V.; Gonzalez, L.; et al. 2023. Human-timescale adaptation in an open-ended task space. In *International Conference on Machine Learning*, 1887–1935. PMLR.

Behymer, K. J.; and Flach, J. M. 2016. From autonomous systems to sociotechnical systems: Designing effective collaborations. *She Ji: The Journal of Design, Economics, and Innovation*, 2(2): 105–114.

Caruana, R. 1997. Multitask learning. *Machine learning*, 28: 41–75.

Chan, A. J.; and van der Schaar, M. 2021. Scalable Bayesian inverse reinforcement learning. *arXiv preprint arXiv:2102.06483*.

Charakorn, R.; Manoonpong, P.; and Dilokthanakul, N. 2022. Generating diverse cooperative agents by learning incompatible policies. In *The Eleventh International Conference on Learning Representations*.

Choi, J.; and Kim, K.-E. 2012. Nonparametric Bayesian inverse reinforcement learning for multiple reward functions. *Advances in neural information processing systems*, 25.

Dafoe, A.; Hughes, E.; Bachrach, Y.; Collins, T.; McKee, K. R.; Leibo, J. Z.; Larson, K.; and Graepel, T. 2020. Open problems in cooperative AI. *arXiv preprint arXiv:2012.08630*.

Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; and Whiteson, S. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Foerster, J.; Nardelli, N.; Farquhar, G.; Afouras, T.; Torr, P. H.; Kohli, P.; and Whiteson, S. 2017. Stabilising experience replay for deep multi-agent reinforcement learning. In *International conference on machine learning*, 1146–1155. PMLR.

Genter, K.; Agmon, N.; and Stone, P. 2011. Role-based ad hoc teamwork. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, 1782–1783.

Gurulingan, N. K.; Zonooz, B.; and Arani, E. 2023. Multi-Task Structural Learning using Local Task Similarity induced Neuron Creation and Removal. *arXiv preprint arXiv:2305.00441*.

Hu, H.; and Sadigh, D. 2023. Language instructed reinforcement learning for human-AI coordination. *arXiv preprint arXiv:2304.07297*.

Iqbal, S.; and Sha, F. 2019. Actor-attention-critic for multi-agent reinforcement learning. In *International conference on machine learning*, 2961–2970. PMLR.

Johnson, M.; Vignatti, M.; and Duran, D. 2020. Understanding human-machine teaming through interdependence analysis. In *Contemporary Research*, 209–233. CRC Press.

Knapp, M. L.; Hall, J. A.; and Horgan, T. G. 2013. *Nonverbal communication in human interaction*. Cengage Learning.

Kraemer, L.; and Banerjee, B. 2016. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190: 82–94.

Li, H.; and He, H. 2023. Multiagent Trust Region Policy Optimization. *IEEE Transactions on Neural Networks and Learning Systems*.

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

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; and Mordatch, I. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *Neural Information Processing Systems (NIPS)*.

Mandyam, A.; Li, D.; Cai, D.; Jones, A.; and Engelhardt, B. 2022. Efficient Bayesian Inverse Reinforcement Learning via Conditional Kernel Density Estimation. In *Fourth Symposium on Advances in Approximate Bayesian Inference*.

Mandyam, A.; Li, D.; Cai, D.; Jones, A.; and Engelhardt, B. E. 2023. Kernel Density Bayesian Inverse Reinforcement Learning. *arXiv preprint arXiv:2303.06827*.

Matignon, L.; Laurent, G. J.; and Le Fort-Piat, N. 2007. Hysteretic q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 64–69. IEEE.

Mirsky, R.; Carlucho, I.; Rahman, A.; Fosong, E.; Macke, W.; Sridharan, M.; Stone, P.; and Albrecht, S. V. 2022. A survey of ad hoc teamwork research. In *European conference on multi-agent systems*, 275–293. Springer.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 1928–1937. PMLR.

Ng, A. Y.; Russell, S.; et al. 2000. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, 2.

Niu, Z.; Zhong, G.; and Yu, H. 2021. A review on the attention mechanism of deep learning. *Neurocomputing*, 452: 48–62.

Oliehoek, F. A.; Amato, C.; et al. 2016. *A concise introduction to decentralized POMDPs*, volume 1. Springer.

Omidshafiei, S.; Pazis, J.; Amato, C.; How, J. P.; and Vian, J. 2017. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *International Conference on Machine Learning*, 2681–2690. PMLR.

Panait, L.; Sullivan, K.; and Luke, S. 2006. Lenient learners in cooperative multiagent systems. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 801–803.

Papoudakis, G.; Christianos, F.; and Albrecht, S. 2021. Agent modelling under partial observability for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 19210–19222.

Ramachandran, D.; and Amir, E. 2007. Bayesian Inverse Reinforcement Learning. In *IJCAI*, volume 7, 2586–2591.

Rashid, T.; Farquhar, G.; Peng, B.; and Whiteson, S. 2020. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 33: 10199–10210.

Robinette, P.; Li, W.; Allen, R.; Howard, A. M.; and Wagner, A. R. 2016. Overtrust of robots in emergency evacuation scenarios. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 101–108. IEEE.

Rusu, A. A.; Rabinowitz, N. C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; and Hadsell, R. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Simmler, M.; and Frischknecht, R. 2021. A taxonomy of human–machine collaboration: Capturing automation and technical autonomy. *Ai & Society*, 36(1): 239–250.

Spaan, M. T. 2012. Partially observable Markov decision processes. In *Reinforcement learning: State-of-the-art*, 387–414. Springer.

Traeger, M. L.; Strohkorb Sebo, S.; Jung, M.; Scassellati, B.; and Christakis, N. A. 2020. Vulnerable robots positively shape human conversational dynamics in a human–robot team. *Proceedings of the National Academy of Sciences*, 117(12): 6370–6375.

Treutlein, J.; Dennis, M.; Oesterheld, C.; and Foerster, J. 2021. A new formalism, method and open issues for zero-shot coordination. In *International Conference on Machine Learning*, 10413–10423. PMLR.

Van der Vaart, A. W. 2000. *Asymptotic statistics*, volume 3. Cambridge university press.

Vezhnevets, A. S.; Osindero, S.; Schaul, T.; Heess, N.; Jaderberg, M.; Silver, D.; and Kavukcuoglu, K. 2017. Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*, 3540–3549. PMLR.

Wang, Y.; Chen, Y.; Jamieson, K.; and Du, S. S. 2023. Improved active multi-task representation learning via lasso. In *International Conference on Machine Learning*, 35548–35578. PMLR.

Watkins, C. J.; and Dayan, P. 1992. Q-learning. *Machine learning*, 8: 279–292.

Weiss, K.; Khoshgoftaar, T. M.; and Wang, D. 2016. A survey of transfer learning. *Journal of Big data*, 3(1): 1–40.

Whiteson, S.; Samvelyan, M.; Rashid, T.; De Witt, C.; Farquhar, G.; Nardelli, N.; Rudner, T.; Hung, C.; Torr, P.; and Foerster, J. 2019. The StarCraft multi-agent challenge. In

Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS, 2186–2188.

Yang, T.; Wang, W.; Tang, H.; Hao, J.; Meng, Z.; Mao, H.; Li, D.; Liu, W.; Chen, Y.; Hu, Y.; et al. 2021. An efficient transfer learning framework for multiagent reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 17037–17048.

Yu, C.; Gao, J.; Liu, W.; Xu, B.; Tang, H.; Yang, J.; Wang, Y.; and Wu, Y. 2023. Learning zero-shot cooperation with humans, assuming humans are biased. *arXiv preprint arXiv:2302.01605*.

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

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

Zhou, H.; Lan, T.; and Aggarwal, V. 2022. PAC: Assisted Value Factorization with Counterfactual Predictions in Multi-Agent Reinforcement Learning. *Advances in Neural Information Processing Systems*, 35: 15757–15769.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; Dey, A. K.; et al. 2008. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, 1433–1438. Chicago, IL, USA.

Zintgraf, L.; Devlin, S.; Ciosek, K.; Whiteson, S.; and Hofmann, K. 2021. Deep interactive bayesian reinforcement learning via meta-learning. *arXiv preprint arXiv:2101.03864*.