A Causality-Inspired Spatial-Temporal Return Decomposition Approach for Multi-Agent Reinforcement Learning

Yudi Zhang[1] **Yali Du**[2] **Biwei Huang**[3] **Meng Fang**[5, 1] **Mykola Pechenizkiy**[1]
[1]Eindhoven University of Technology [2]King's College London
[3]University of California San Diego [5]University of Liverpool
{y.zhang5,m.pechenizkiy}@tue.nl, yali.du@kcl.ac.uk
bih007@ucsd.edu, Meng.Fang@liverpool.ac.uk

Abstract

Multi-agent reinforcement learning (MARL) has been largely developed to solve multi-agent cooperation problems. However, it remains insufficiently developed for explaining decision-making processes. This challenge becomes particularly pronounced with delayed rewards, especially episodic ones, as credit allocation must be accounted for along both the temporal and spatial axes involving multiple agents. In this paper, we propose a **CA**usally-inspired **S**patial-**T**emporal return decomposition method, named **CAST**, to tackle episodic reward in cooperative MARL. We provide interpretable return decomposition and allow the complexity of multi-agent dynamics by relaxing the common assumption. Specifically, along the temporal dimension, episodic long-term return satisfies a linear summation of team rewards from all time steps. More interestingly, in the spatial dimension, beyond a simple linear summation of individual rewards, team rewards are allowed to be general nonlinear mixtures of individual rewards, facilitating more reasonable and precise credit allocation. We theoretically show that, under the proposed framework, the team rewards, individual rewards, and underlying causal relationships are identifiable, which naturally introduces additional structure constraints to enhance the interpretability of reward redistribution. Our experiments demonstrate state-of-the-art results in MPE and its variants, and the provided visualization of the causal structure demonstrates the interpretability of our method.

1 Introduction

Cooperative Multi-agent Reinforcement Learning (MARL) is a burgeoning area that allows agents to learn to collaborate towards a shared team goal [24, 49, 48, 45]. It widely accelerates multiple applications of AI in the real world, such as games [38, 44, 30] and robotics [17, 33]. The challenges of learning individual cooperative policies stem not only from the complex dynamics where each agent's actions influence both their own observations and those of others but also from the single scalar team reward, which measures their collective performance. These factors pose the central challenge in cooperative MARL: credit assignment [28, 34, 39, 15, 7]. Those methods work well while the team reward signals are dense. However, a more practical but challenging scenario is where the team of agents is only awarded sparse and delayed rewards at the end of the episode. Recently, works along spatial-temporal credit assignment propose to explicitly redistribute the individual rewards for the agents and timesteps to mitigate this problem [32, 4].

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

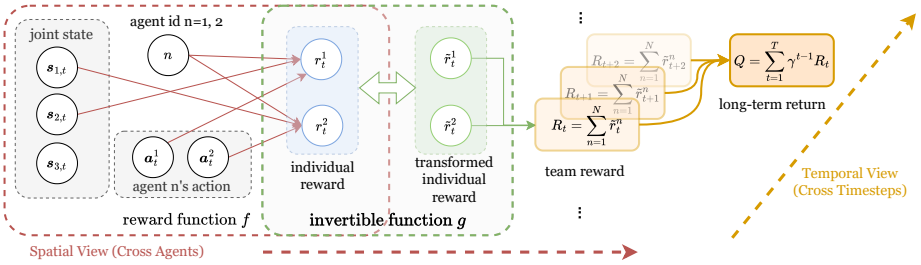

Figure 1: A graphical illustration $\mathcal{G}$ of the causal process that describes the generation of long-term return in general. Individual reward $\{r_t^1 \cdots, r_t^N\}$ is generated from the joint state $\{\boldsymbol{s}_{1,t} \cdots \boldsymbol{s}_{|\boldsymbol{s}|,t}\}$ and joint action $\{\boldsymbol{a}_t^1 \cdots, \boldsymbol{a}_t^N\}$ by reward function f, and transformed individual reward $\{\tilde{r}_t^1, \cdots \tilde{r}_t^N\}$ is generated from individual reward by invertible function g. Accordingly, team reward R_t is the causal effect of individual rewards of all agents, and long-term reward Q equals the sum of team rewards from all the timesteps.

Previous work often lacks detailed interpretability about why specific contributions are generated. Understanding how states and actions impact the team's outcome can provide valuable insights for further policy learning. Some previous work emphasizes the benefits of utilizing the structural generation process of rewards. For instance, AdaRL constructs a compact representation for policy learning [11], and LAIES[16] uses given structural information to design intrinsic rewards. However, achieving both reasonable and precise credit allocation under the complex dynamics of MARL while also providing interpretability is not trivial. Current studies commonly decompose the long-term return into individual rewards through a simple structural assumption by assuming the team reward is a direct summation of individual rewards [4, 32]. Although they do not explicitly estimate the causal structures behind the individual reward generation, such a simple and strict summation assumption indeed can guarantee an identifiable causal structure for allocating individual credits. However, it oversimplifies the complex dynamics in MARL and may hinder the ability to allocate credit accurately and reasonably. Let's consider a team game scenario with three players: A, B, and C. Players A and B are responsible for attacking the enemy, while player C is a medic who heals A and B, allowing them to survive longer and deal more damage over time. It is not reasonable to simply equal the success of the game to the damage dealt by A and B, ignoring the essential support provided by C, which is usually complex, in enhancing the overall performance of the team. Hence, the credit assignment in such a case is beyond linear decomposition and instead an unknown mixture of agents' individual contributions, where using the linear summation assumption to estimate individual reward functions can limit the representation capability of the learned reward predictor, thereby impairing reasonable and precise credit allocation.

In this paper, we propose a **CA**usality-inspired **S**patial-**T**emporal return decomposition (**CAST**) method, providing the interpretability of allocating agents' individual contributions. More specifically, as shown in Figure 1, along the temporal dimension, CAST models the long-term return as the causal effect of team rewards at all the timesteps, enabling decomposition of the long-term return into a team reward for each timestep [29, 47, 1]; Along the spatial dimension, CAST views team reward as the causal effect of the actual individual contributions (for simplicity, we use individual rewards in the rest of the paper) of all the agents. What makes CAST advance beyond previous work [4] is its departure from the simple linear assumption, *i.e.*, equating the team reward to the sum of agents' individual rewards. Instead, CAST adopts a more general and flexible modeling where the team reward is generated from a nonlinear mixture of individual rewards, inspired by iVAE [14]. We assume an invertible mixture function to map the individual rewards to immediate transformed rewards, which sum to the team reward, thus guaranteeing the identifiability of the causal structures and the unobserved individual reward functions within the generative process of multi-agent systems. Such identifiability further enables us to incorporate explicit structural modeling and constraints that provide interpretability. Overall, we not only provide interpretability but also relax the assumption of additive linear team reward as a nonlinear mixture of individual rewards which allows a much richer class of possible functions.

Our contributions are four-faceted. (1) We expand the theoretical understanding of spatial-temporal credit assignment in MARL by introducing a nonlinear invertible mixture model for team reward gen-

eration. The proposed method, CAST, captures the complex dynamics and ensures the identifiability of causal relationships and unobserved individual reward functions within the generative process of cooperative MARL. (2) We explicitly estimate causal relationships that influence the generation of individual rewards, thereby modeling the contribution of agents and providing interpretability. (3) We design an individual reward predictor based on iVAE [14] under the nonlinear mixture assumption. (4) Experimental results on the classic and modified Multi-agent Particle environment underscore the superiority of our approach, and the visualization verifies the interpretability of our method.

2 CAST: Causality-Inspired Spatial-Temporal Return Decomposition

We focus on enhancing policy learning through explicit credit assignments in cooperative games with sparse and delayed team rewards, especially episodic ones. We begin by describing the generative process in cooperative games, which lays the foundation for our proposed approach. We then detail a technique for recovering this process and conducting policy learning using individually assigned rewards. Similar to the previous work [47], the proposed method, **CAST**, contains a generative model Φ_{m} and a policy model Φ_π. After learning individual reward functions for each agent through generative model learning, we can exploit the predicted individual rewards for independent policy learning. The overall learning objective is to minimize,

$$L(\Phi_{\mathrm{m}}, \Phi_\pi) = L_{\mathrm{m}}(\Phi_{\mathrm{m}}) + J_\pi(\Phi_\pi), \tag{1}$$

where the L_{m} is designed for model estimation (defined in Eq. 4) and J_π is for policy learning depends on specific RL optimization algorithm (defined in Eq. 6).

2.1 Underlying Generative Process in MARL

We first state the setting of CAST: for each timestep t with a joint state $\boldsymbol{s}_t$, the agent $n \in [1, \cdots, A]$ takes action $\boldsymbol{a}_t^n$, forming the joint action $\boldsymbol{a}_t$, and contributes to the team reward R_t by individual reward r_t^n (the goal of spatial-temporal credit assignment). However, the agent can only observe an episodic reward, which measures the performance of the entire trajectory at $t = T$; otherwise, the agent receives zero rewards.

Generative Model. As shown in Figure 1, CAST exploits a Dynamic Bayesian Network (DBN) [22], $\mathcal{G}$, over a finite number of random variables,

$$\{\underbrace{\{\boldsymbol{s}_{1,t}, \cdots, \boldsymbol{s}_{|\boldsymbol{s}|,t}\}}_{\text{joint state}} \cup \underbrace{\{\boldsymbol{a}_t^1, \cdots \boldsymbol{a}_t^N\}}_{\text{agents' action}} \cup \underbrace{\{r_t^1, \cdots, r_t^N\}}_{\text{individual rewards}} \cup \underbrace{\{\tilde{r}_t^1, \cdots, \tilde{r}_t^N\}}_{\text{transformed individual rewards}} \}_{t=1}^{T} \bigcup Q,$$

where $|\boldsymbol{s}|$ and $|\boldsymbol{a}^n|$ are the dimension of $\boldsymbol{s}_t$ and $\boldsymbol{a}_t^n$, N is the number of agents, and $\mathcal{G}$ characterizes the underlying generative process in MARL as follows,

$$\begin{cases} \text{individual reward:} & r_t^n = f^n(\boldsymbol{C}^n \odot \boldsymbol{s}_t, \boldsymbol{a}_t^n, \epsilon_{r,n,t}) \\ \text{team reward:} & R_t = \sum_{n=1}^{N} g^n([r_t^1, r_t^2, \cdots, r_t^N]) \\ \text{long-term return:} & Q = \sum_{t=1}^{T} \gamma^{t-1} R_t \end{cases} \tag{2}$$

where $\odot$ is the element-wise product. Note that such causal modeling relaxes the previous strict linear assumption [4], which can be regarded as a special case of our model when g is an identity function. In our experimental environments, where the state is not available during training, we use the agents' observations $\boldsymbol{o}_t$ as a proxy for the environmental state $\boldsymbol{s}_t$ and agents' index n.

Notations. For simplicity, we denote $\tilde{r}_t^n = g^n(\boldsymbol{r}_t)$, $\boldsymbol{r}_t = [r_t^1, \cdots, r_t^N]$ and $\tilde{\boldsymbol{r}}_t = [\tilde{r}_t^1, \cdots, \tilde{r}_t^N]$. We denote by r_t^n the individual reward at time step t of agent n. In the rest of the paper, we call $\tilde{r}_t^n$ as *transformed individual rewards*. Q is the trajectory-wise long-term return. T is the maximum episode length of the environment. $\epsilon_{r,n,t}$ is the *i.i.d.* noise.

Causal structure and interpretability. $\boldsymbol{C}^n, \forall n \in [1, \cdots, N]$ is a binary mask to capture the causal structure between the elements of joint state and individual rewards of agents, with $\boldsymbol{C}^n \in \{0, 1\}^{|\boldsymbol{s}|}$. $\boldsymbol{C}^n$ controls if a specific dimension of the state $\boldsymbol{s}_t$ impacts the individual reward r_t^n at timestep t. Let $\boldsymbol{C}_k^n$ be the k-th element in the vector $\boldsymbol{C}^n$. If there is an edge from the k-th dimension of $\boldsymbol{s}_t$ to the agent n's individual reward r_t^n in $\mathcal{G}$, then $\boldsymbol{C}_k^n = 1$. Given $\mathcal{G}$, we can naturally explain how the individual rewards are generated, *i.e.*, the explicit contribution of each dimension of the joint state towards individual rewards.

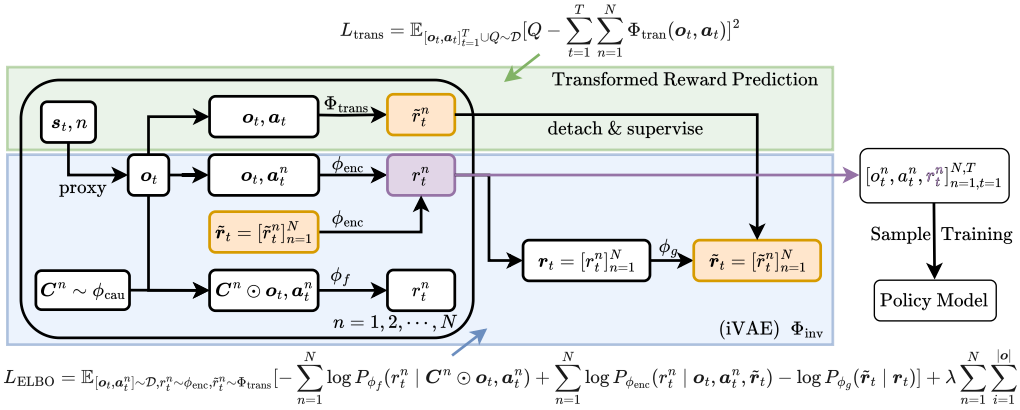

Figure 2: The overall pipeline of the proposed method. **Generative Model $\Phi_{\mathbf{m}}$:** The causal structures for prediction of individual rewards r_t^n are $\boldsymbol{C}_t^n$, sampled from ϕ_{cau}; The module (Φ_{trans}, in green) takes $(\boldsymbol{o}_t, \boldsymbol{a}_t^n)$ as input to predict the transformed individual rewards $\tilde{r}_t^n$ (in orange) and optimized by L_{trans}. The module (Φ_{inv}, in blue) is constructed as an iVAE, consisting of an encoder (ϕ_{enc}), a decoder (ϕ_{g}), an individual reward predictor (ϕ_f). The encoder takes as input $\boldsymbol{o}_t$, $\boldsymbol{a}_t^n$ to predict the individual rewards r_t^n, individual rewards $\tilde{\boldsymbol{r}}_t$. We treat the latent of the iVAE as the individual rewards r_t^n (in purple) and optimize Φ_{inv} through L_{ELBO}. **Policy Φ_π:** The prediction of r_t^n from the encoder ϕ_{enc} finally guides the independent policy learning process.

Functions in Eq. 2. f is the unknown individual reward function, whose output r_t^n is expected to accurately describe the contribution of agent n, and serves as the reward signals for independent policy learning. g is an invertible function to generate the transformed individual rewards $\tilde{\boldsymbol{r}}_t$ from individual rewards $\boldsymbol{r}_t$, *i.e.*, $\tilde{\boldsymbol{r}}_t$ is a nonlinear or linear invertible mixture of $\boldsymbol{r}_t$. We assume that the sum of transformed individual rewards $\sum_{n=1}^{N} \tilde{r}_t^n$ equals Q, where we follow previous work to ignore discount factor γ [29].

Relaxed assumption. We want to highlight the complexity of team reward generation in a multi-agent system, which can range from simple linear sums to complex nonlinear functions. This contrasts with previous work, such as STAS [4], which assumes that the team reward equals the sum of individual rewards. Such a restrictive assumption limits the reasonable and precise credit assignment in the complex MARL system. In contrast, our proposed framework allows the team reward to be a general mixture of individual rewards r_t^n, with the linear assumption being a special case when $\boldsymbol{r}_t = \tilde{\boldsymbol{r}}_t$.

2.2 Theoretical Results

In this subsection, we provide 1) the identifiability results of the unknown functions and structures in Eq. 2, which support the estimation of the causal structure from the data, enabling the interpretability of our method; 2) the equivalence of using the decomposed rewards for policy learning in our proposed framework.

Proposition 2.1 (Identifiability for Spatial-Temporal Credit Assignment)**.** *Consider the data generating process in Eq. 2. Suppose the joint state $\boldsymbol{s}_t$, the action $\boldsymbol{a}_t^n$ for each agent n and the long-term return (can be calculated by the discounted sum of delayed rewards) are observable, while the individual r_t^n for each agent n and team reward R_t are unobserved.*

Under the Markov condition (Definition E.2) and faithfulness assumption (Definition E.3), if the function g for generating the transformed individual rewards is invertible, then the causal mask $\boldsymbol{C}^n$ is identifiable and we can identify the individual rewards r_t^n to their monolithic invertible transformations, e.g. $\log(r_t^n)$.

Proof Sketch. The proof begins by establishing the identifiability of the transformed individual rewards, represented by $\tilde{r}^n$, indicating the possibility of recovering it from the data. The second part of the proof highlights the relationship between our method and nonlinear Independent Component Analysis (ICA), along with confirming the identifiability of individual rewards, r_t^n. □

Remark 2.2. *The proposition 2.1 shows that we can identify causal structures and individual rewards (up to their invertible nonlinear transformation) from the observed data. The proof is in Appendix F.*

Based on Proposition 2.1, since the individual rewards are 1-dimensional, it is equivalently to say that we can identify the individual rewards up to their monolithic function of the ground truth individual rewards. Since the recovered individual rewards can be positively or negatively correlated to their ground truth, we give the Proposition 2.3 for policy learning with the estimated individual rewards.

Proposition 2.3. *If $k(\cdot)$ is a monotonically increasing invertible transformation, then it is equivalence to optimize the policy using the ground individual rewards r_t^n and its k-based transformation $k(r_t^n)$.*

2.3 Generative Model Learning

In this subsection, we present how our proposed method recovers the underlying generative process. This includes the identification of binary masks ($\boldsymbol{C}$), as well as the estimation of unknown functions (f and g). The parameterized model Φ_m that incorporates the structures and functions in Eq. 2 is used to approximate the causal generative process. The pseudocode is in Pseudocode 1 in Appendix I.

Generative Model. As shown in Figure 2, Φ_{m} consists of two parts: (1) Φ_{trans} for prediction of transformed rewards $\tilde{r}_t^n = \Phi_{\text{trans}}(\boldsymbol{o}_t, \boldsymbol{a}_t^n)$; (2)$\Phi_{\text{inv}} := [\phi_{\text{cau}}, \phi_f, \phi_{\text{enc}}, \phi_g]$ for prediction of individual rewards r_t^n while keep the function g invertible.

We illustrate more details about Φ_{inv} as follows. We first exploit a set of free parameters, $\phi_{\text{cau}} \in \mathbb{R}^{N\times|\boldsymbol{s}|}$, to estimate the existence of the causal edges for the generation of the individual rewards r_t^n. Define $\phi_{i,j,\text{cau}}^n$ as the (n,i,j)-th element of ϕ_{cau}, where $n \in [1,N], i \in [1,|\boldsymbol{o}^i|], j \in [0,1]$. The existence of the edge $\boldsymbol{C}_j^n, \forall n \in [1,N], j \in [1,|\boldsymbol{o}^i|]$, is modeled by $\phi_{i,j,cau}^n$ whose value fails in between 0 and 1. Then, given the output of Φ_{trans}, the transformed individual rewards $\tilde{r}_t^n$, we construct an iVAE [14] to mimic the invertible function, which consists of $\phi_f, \phi_{\text{enc}}, \phi_g$. It is defined as follows,

$$\text{Prior: } r_t^n = \phi_f(\boldsymbol{C}^n \odot \boldsymbol{o}_t^n, \boldsymbol{a}_t^n), \text{ Encoder: } r_t^n = \phi_{\text{enc}}(\boldsymbol{C}^n \odot \boldsymbol{o}_t^n, \boldsymbol{a}_t^n, \tilde{\boldsymbol{r}}_t), \text{ Decoder: } \tilde{\boldsymbol{r}}_t = \phi_g(\boldsymbol{r}_t), \quad (3)$$

where $\tilde{\boldsymbol{r}}_t = [\tilde{r}_t^1, \cdots, \tilde{r}_t^N]$ and $\boldsymbol{r}_t = [r_t^1, \cdots, r_t^N]$. In Eq. 3, ϕ_f and ϕ_{enc} both aim to predict the individual rewards. While both of them take as input the causal structure, the joint state, as well as the action and index of an individual agent, the latter one takes one more, *i.e.*, the prediction of transformed individual reward $\tilde{r}_t^n$. Hence, we consider that the prediction of r_t^n from ϕ_{enc} is more informative and use it for further policy learning.

Learning Objective. The overall loss term for generative model learning is as follows,

$$L_{\text{m}}(\Phi_{\text{m}}) = L_{\text{trans}}(\Phi_{\text{trans}}) + L_{\text{ELBO}}(\Phi_{\text{inv}}). \quad (4)$$

Specifically, L_{trans} is responsible for optimizing Φ_{trans} to predict the transformed individual rewards: $L_{\text{trans}} = \mathbb{E}_{\tau\sim\mathcal{D}}||Q - \sum_{t=1}^{T}\sum_{n=1}^{N}\Phi_{\text{trans}}(\boldsymbol{o}_t^n, \boldsymbol{a}_t)||^2$. where $\tau := [\boldsymbol{o}_t, \boldsymbol{a}_t]_{t=1}^T \cup Q$. For the optimization of Φ_{inv}, we maximize the lower bound on the log-likelihood and define L_{ELBO} as follows,

$$\begin{aligned} L_{\text{ELBO}} \quad &= \mathbb{E}_{\boldsymbol{r}_t\sim\phi_{\text{enc}}, \tilde{\boldsymbol{r}}_t\sim\Phi_{\text{trans}}, \tau\sim\mathcal{D}}\Big[-\sum_{n=1}^{N}\log P_{\phi_f}(r_t^n \mid \boldsymbol{C}^n \odot \boldsymbol{o}_t^n, \boldsymbol{a}_t^n) \\ &\quad + \sum_{n=1}^{N}\log P_{\phi_{\text{enc}}}(r_t^n \mid \boldsymbol{o}_t^n, \boldsymbol{a}_t^n, \tilde{\boldsymbol{r}}_t) - \log P_{\phi_g}(\tilde{\boldsymbol{r}}_t \mid \boldsymbol{r}_t)\Big] + \lambda\sum_{n=1}^{N}\sum_{i=1}^{|\boldsymbol{o}^i|}\boldsymbol{C}_i^n, \end{aligned} \quad (5)$$

where $\tau = [\boldsymbol{o}_t, [\boldsymbol{a}_t^n]_{n=1}^N]_{t=1}^T$, $\tilde{r}_t^n$ is generated by Φ_{trans}, dropping the gradients, and the last loss term is for regulating the sparsity of learned causal structure, avoiding trial solutions. For more details about the model structure and hyper-parameters, please refer to Appendix I.

2.4 Policy Learning

After assigning the individual rewards r_t^n predicted by ϕ_{enc}, we convert the multi-agent learning into independent single-agent policy training. In the experiments, we adopt Proximal Policy Optimization (PPO) [31] for independent policy optimization and let all the agents share the same policy. The policy model Φ_π contains two parts, a critic ϕ_v and an actor ϕ_π due to the applied PPO algorithm. PPO trains an actor $\phi_\pi(\boldsymbol{o}_t^n)$ by minimizing,

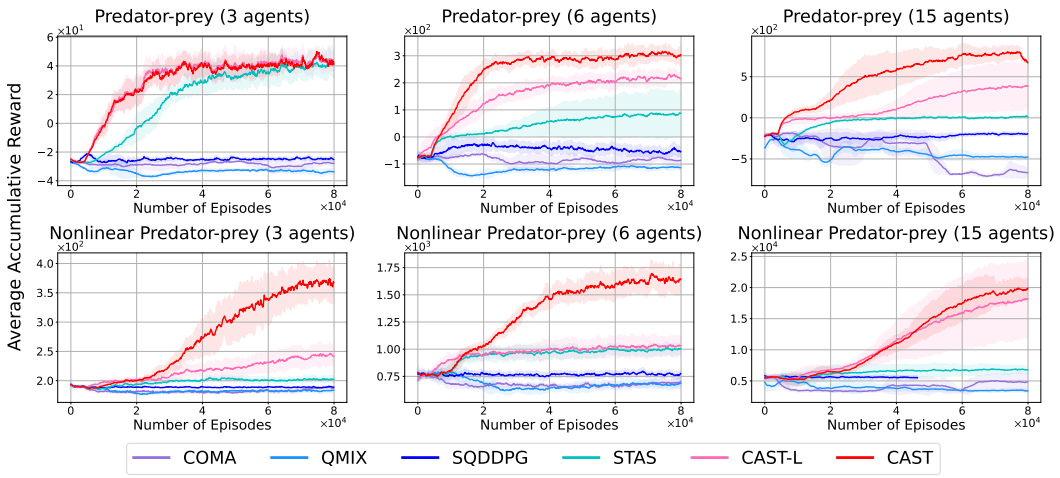

Figure 3: Learning curve on Multi-agent Particle Environment with episodic rewards, based on 3 independent runs with random initialization. The shaded region indicates the standard deviation and the lines are smoothed by averaging the 10 most recent evaluation points using an exponential moving average.

$$J_\pi = \mathbb{E}\left[\min\left(\frac{\pi_\phi(\boldsymbol{a}_t^n \mid \boldsymbol{o}_t^n)}{\phi_{\phi,\text{old}}(\boldsymbol{a}_t^n \mid \boldsymbol{o}_t^n)}\hat{A}_t, \text{clip}\left(\frac{\pi_\phi(\boldsymbol{a}_t^n \mid \boldsymbol{o}_t^n)}{\phi_{\phi,\text{old}}(\boldsymbol{a}_t^n \mid \boldsymbol{o}_t^n)}, 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right] \tag{6}$$

where $\hat{A}$ is the advantage function in PPO [31]. Detailed implementation of Φ_π is in Appendix I.

Search the positive correlated individual rewards. Supported by the identifiability result, the proposed method is able to identify the individual rewards up to its monolithic transformation. However, it is unclear whether the recovered reward is positively or negatively correlated with the actual agent's contribution (individual rewards). Consequently, given the recovered individual reward r, we applied policy learning for both the positive reward (r) and negative reward ($-r$), and then selected the policy that demonstrated better performance. According to Proposition 2.3, we can learn the optimal policy using the recovered positive correlated nonlinear invertible transformation of the actual individual rewards.

3 Experiments

In this section, we begin with evaluating our method in the Multi-agent Particle Environment (MPE) and its variants against several baselines. Then, we conduct ablation studies to verify the benefit of relaxing the linear assumption and investigate the interpretability of our methods.

Setup We begin with the setup of our experiments, including the baselines, environments, and metrics. **Baselines.** We compare our method with four baselines, including COMA [7], QMIX [28], SQDDPG [41] and STAS [4]. The implementation of baselines is from STAS [4]. For more details, please refer to Appendix G.2. **Environments.** We evaluate our method and investigate its reasonable design in a Multi-agent Particle Environment (MPE) [21, 19]. More specifically, we utilize the challenging episodic scenarios built upon the classic implementation of MPE, *Classical MPE (Episodic)* and the variant of Episodic MPE with nonlinear team rewards, named *Nonlinear MPE (Episodic)*. For more details, please refer to Appendix I

3.1 Main Results

We provide the learning curve on the classic and modified MPE scenarios, as shown in Figure 3. The line in red and pink denote our method using the recovered individual rewards (**CAST**) and the recovered transformed individual rewards, (**CAST-L**) separately. The others are baseline methods. The proposed method achieves a higher cumulative reward than the others across most different scenarios in the Multi-agent Particle Environment (MPE), including the classic MPE scenarios (linear, first row) and the modified MPE (nonlinear, last row). Among the baselines, STAS [4] explicitly addresses spatial-temporal credit assignment by reward redistribution thus obtaining good performance in the classic MPE (first row). However, limited by the assumption of a linear sum of

individual rewards, they fail to credit the agents by a granular contribution allocation in non-linear MPE. This observation demonstrates that applying a simple linear assumption to credit the agents is not suitable in the nonlinear team reward setting, which is a more general setting in the real world. In contrast, although **CAST** relaxes the linear assumption, it can perform well in both linear and nonlinear team reward settings, revealing the generalizability of our method.

3.2 Ablation Study

We conduct the following ablation study: 1) the results of the ablation versions of our method: **CAST-L** decomposes rewards under linear assignment **CA-T**: policy learning with predicted team rewards, *i.e*. without spatial credit assignment; 2) the results of using ground truth rewards for policy learning: methods named **GT (team), GT (trans), GT** use ground truth team reward, transformed individual rewards and individual rewards, separately for policy learning; 3) the visualization of causal structure to demonstrate the interpretability of our method; 4) the Spearman's rank correlation coefficient of the recovered individual rewards with the ground truth.

Policy learning with Linear Modeling. To demonstrate the necessity of relaxing the strict linear team reward assumption, we provide the experimental results of policy with predicted transformed individual rewards **CAST-L**, which can be treated as the redistributed reward under the linear assumption. **CAST-L** obtains a good performance in the classic MPE while failing in the nonlinear MPE, which demonstrates that through relaxing the linear assumption in previous work, our method achieves a better performance.

Policy Learning without Spatial Credit Assignment. We provide the experimental results of **CA-T**, using team rewards (sum of predicted transformed individual rewards) for policy learning, *i.e*. without spatial credit assignment. According to Figure A2, lacking explicit spatial credit assignment, the contribution of individual agents to the overall team performance is not distinctly identified, inducing worse performance than that under spatial-temporal credit assignment.

Visualization of Causal Structure on MPE. Here, we delve into the visualization of the learned causal structure in the MPE, the scenarios *Predator-Prey (3 agents)* and *Predator-Prey (6 agents)* to highlight the interpretability of our method by understanding the underlying causal relationships between agents' state and their outcomes. As shown in Figure 4, the lighter the color, the higher the edge exists. According to the reward design in the scenarios, the agent's individual reward is decided by their minimal distance from the prey. Therefore, the causal edge only exists from the prey's relative position (P4 in Figure 4) to the individual rewards, which is consistent with the learned causal structure.

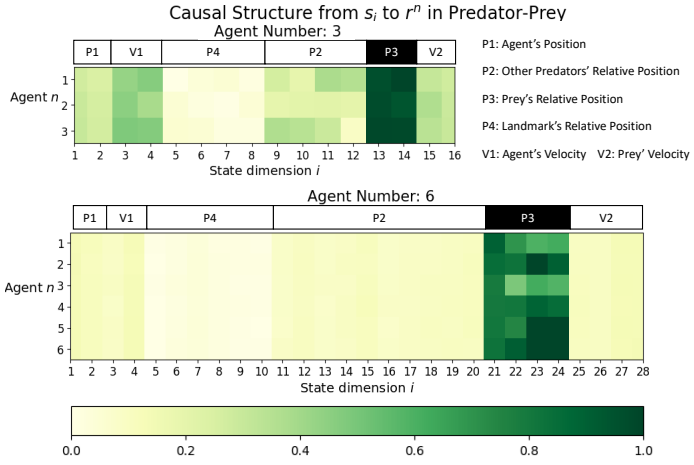

Figure 4: Learned causal structure from i-th dimension of observation to individual reward of agent n in *Predator-Prey (3/6 agents)*. The darker, the higher the probability of the causal edges existing. The black block denotes the ground truth causal structure.

Additional Experimental Results. We provide more experimental results of policy learning with ground truth rewards and the recovery accuracy of the individual rewards in Appendix H.

4 Conclusion

In conclusion, our paper addresses the challenge of credit assignment in cooperative multi-agent learning, specifically focusing on delayed rewards in episodic scenarios. By modeling reward generation along spatial and temporal axes causally, we introduce a novel framework, CAST, to address interpretable spatial-temporal credit assignment. We relax the conventional assumption

of equal values between the sum of individual contributions to team outcomes and propose more general causal modeling to allow the complex generation of the team reward. We show its state-of-the-art results in MPE and its variants and verify the method's interpretability through insightful visualizations.

References

[1] J. A. Arjona-Medina, M. Gillhofer, M. Widrich, T. Unterthiner, J. Brandstetter, and S. Hochreiter. Rudder: Return decomposition for delayed rewards. *Advances in Neural Information Processing Systems*, 32, 2019.

[2] A. Ben-Israel. The change-of-variables formula using matrix volume. *SIAM Journal on Matrix Analysis and Applications*, 21(1):300–312, 1999.

[3] Y.-H. Chang, T. Ho, and L. Kaelbling. All learning is local: Multi-agent learning in global reward games. *Advances in neural information processing systems*, 16, 2003.

[4] S. Chen, Z. Zhang, Y. Du, and Y. Yang. Stas: Spatial-temporal return decomposition for multi-agent reinforcement learning. *arXiv preprint arXiv:2304.07520*, 2023.

[5] C. S. de Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. Torr, M. Sun, and S. Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.

[6] F. Feng, B. Huang, K. Zhang, and S. Magliacane. Factored adaptation for non-stationary reinforcement learning. *Advances in Neural Information Processing Systems*, 2022.

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

[8] S. J. Grimbly, J. Shock, and A. Pretorius. Causal multi-agent reinforcement learning: Review and open problems. In *Cooperative AI Workshop, Advances in Neural Information Processing Systems*, 2021.

[9] H. Hu, Y. Yang, J. Ye, Z. Mai, and C. Zhang. Unsupervised behavior extraction via random intent priors. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[10] J. Hu, P. Stone, and R. Martín-Martín. Causal policy gradient for whole-body mobile manipulation, 2023.

[11] B. Huang, F. Feng, C. Lu, S. Magliacane, and K. Zhang. Adarl: What, where, and how to adapt in transfer reinforcement learning. *CoRR*, abs/2107.02729, 2021.

[12] B. Huang, F. Feng, C. Lu, S. Magliacane, and K. Zhang. Adarl: What, where, and how to adapt in transfer reinforcement learning. In *International Conference on Learning Representations*, 2021.

[13] N. Jaques, A. Lazaridou, E. Hughes, C. Gulcehre, P. Ortega, D. Strouse, J. Z. Leibo, and N. De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International conference on machine learning*, pages 3040–3049. PMLR, 2019.

[14] I. Khemakhem, D. Kingma, R. Monti, and A. Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020.

[15] J. Li, K. Kuang, B. Wang, F. Liu, L. Chen, F. Wu, and J. Xiao. Shapley counterfactual credits for multi-agent reinforcement learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 934–942, 2021.

[16] B. Liu, Z. Pu, Y. Pan, J. Yi, Y. Liang, and D. Zhang. Lazy agents: A new perspective on solving sparse reward problem in multi-agent reinforcement learning. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 21937–21950. PMLR, 23–29 Jul 2023.

[17] B. Liu, L. Wang, and M. Liu. Lifelong federated reinforcement learning: a learning architecture for navigation in cloud robotic systems. *IEEE Robotics and Automation Letters*, 4(4):4555–4562, 2019.

[18] Y.-R. Liu, B. Huang, Z. Zhu, H. Tian, M. Gong, Y. Yu, and K. Zhang. Learning world models with identifiable factorization. *arXiv preprint arXiv:2306.06561*, 2023.

[19] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Neural Information Processing Systems (NIPS)*, 2017.

[20] X. Lyu, Y. Xiao, B. Daley, and C. Amato. Contrasting centralized and decentralized critics in multi-agent reinforcement learning. *arXiv preprint arXiv:2102.04402*, 2021.

[21] I. Mordatch and P. Abbeel. Emergence of grounded compositional language in multi-agent populations. *arXiv preprint arXiv:1703.04908*, 2017.

[22] K. P. Murphy. *Dynamic bayesian networks: representation, inference and learning*. University of California, Berkeley, 2002.

[23] F. A. Oliehoek and C. Amato. A concise introduction to decentralized pomdps. In *SpringerBriefs in Intelligent Systems*, 2016.

[24] A. Oroojlooy and D. Hajinezhad. A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence*, 53(11):13677–13722, 2023.

[25] V. P. Patil, M. Hofmarcher, M.-C. Dinu, M. Dorfer, P. M. Blies, J. Brandstetter, J. A. Arjona-Medina, and S. Hochreiter. Align-rudder: Learning from few demonstrations by reward redistribution. In *International Conference on Machine Learning*. PMLR, 2022.

[26] J. Pearl. Causality: Models, reasoning, and inference, 2000.

[27] S. Pitis, E. Creager, A. Mandlekar, and A. Garg. Mocoda: Model-based counterfactual data augmentation. *Advances in Neural Information Processing Systems*, 2022.

[28] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *The Journal of Machine Learning Research*, 21(1):7234–7284, 2020.

[29] Z. Ren, R. Guo, Y. Zhou, and J. Peng. Learning long-term reward redistribution via randomized return decomposition. In *International Conference on Learning Representations*, 2022.

[30] M. Samvelyan, T. Rashid, C. S. de Witt, G. Farquhar, N. Nardelli, T. G. J. Rudner, C.-M. Hung, P. H. S. Torr, J. Foerster, and S. Whiteson. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043, 2019.

[31] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[32] J. She, J. K. Gupta, and M. J. Kochenderfer. Agent-time attention for sparse rewards multi-agent reinforcement learning. *arXiv preprint arXiv:2210.17540*, 2022.

[33] D. Shishika, J. Paulos, and V. Kumar. Cooperative team strategies for multi-player perimeter-defense games. *IEEE Robotics and Automation Letters*, 5(2):2738–2745, 2020.

[34] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*, pages 5887–5896. PMLR, 2019.

[35] P. Spirtes, C. N. Glymour, R. Scheines, and D. Heckerman. *Causation, prediction, and search*. MIT press, 2000.

[36] B. K. Sriperumbudur, K. Fukumizu, A. Gretton, A. Hyvärinen, and R. Kumar. Density estimation in infinite dimensional exponential families. *J. Mach. Learn. Res.*, 18:57:1–57:59, 2013.

[37] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, et al. 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*, pages 2085–2087, 2018.

[38] O. Vinyals, T. Ewalds, S. Bartunov, P. Georgiev, A. S. Vezhnevets, M. Yeo, A. Makhzani, H. Küttler, J. Agapiou, J. Schrittwieser, et al. Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*, 2017.

[39] J. Wang, Z. Ren, T. Liu, Y. Yu, and C. Zhang. {QPLEX}: Duplex dueling multi-agent q-learning. In *International Conference on Learning Representations*, 2021.

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

[41] J. Wang, Y. Zhang, T.-K. Kim, and Y. Gu. Shapley q-value: A local reward approach to solve global reward games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7285–7292, 2020.

[42] Z. Wang, Y. Du, Y. Zhang, M. Fang, and B. Huang. Macca: Offline multi-agent reinforcement learning with causal credit assignment. *arXiv preprint arXiv:2312.03644*, 2023.

[43] C. K. Williams and C. E. Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[44] D. Ye, G. Chen, W. Zhang, S. Chen, B. Yuan, B. Liu, J. Chen, Z. Liu, F. Qiu, H. Yu, et al. Towards playing full moba games with deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33:621–632, 2020.

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

[46] J. Zhang and E. Bareinboim. Markov decision processes with unobserved confounders: A causal approach. Technical report, Technical report, Technical Report R-23, Purdue AI Lab, 2016.

[47] Y. Zhang, Y. Du, B. Huang, Z. Wang, J. Wang, M. Fang, and M. Pechenizkiy. Interpretable reward redistribution in reinforcement learning: A causal approach. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[48] M. Zhou, Z. Liu, P. Sui, Y. Li, and Y. Y. Chung. Learning implicit credit assignment for cooperative multi-agent reinforcement learning. *Advances in neural information processing systems*, 33:11853–11864, 2020.

[49] R. Zohar, S. Mannor, and G. Tennenholtz. Locality matters: A scalable value decomposition approach for cooperative multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9278–9285, 2022.

A Related Work

Below we review the related works on credit assignment in multi-agent reinforcement learning, including those on spatial-temporal credit assignment and causality-facilitated reinforcement learning.

Credit assignment in multi-agent reinforcement learning investigates the contribution of each individual agent towards the common goal of the team (measured by a team reward) in cooperative multi-agent environments [3]. A rich line of work [37, 7, 41, 28, 15] focus on value decomposition under the centralized training with decentralized execution (CTDE) paradigm. SHAQ [40] and SQDDPG [41] utilize an agent's approximate Shapley value for credit assignment. Another line of work follows independent learning and demonstrates a robust performance while using decentralized training [5, 20]. Among them, [42] decomposes the team reward and assumes that the team reward equals the sum of contributions of individuals, which we relax in our method. Those methods may not work when the reward signal is extremely sparse and delayed.

Facing the challenge of sparse and delayed rewards in the multi-agent tasks, there is a recent rising need to do Spatial-Temporal Credit Assignment [32, 4]. This line of work explores reward redistribution methods that can be integrated with independent policy learning. ATA [32] extends the RUDDER [1] in a single-agent setting into the multi-agent setting and treats the difference in long-term return predictions between two timesteps as the agent's redistributed rewards. However, the RUDDER-manner redistribution method may not work well in complex environments and lacks interpretability [25, 29, 47]. STAS [4] works in a similar way to us to learn the individual reward redistribution model for each agent, which is more flexible. After the decomposition of the long-term return into each timestep, STAS uses Shapley values to redistribute the individual payoff of agents. However, STAS assumes that the long-term return is equal to the linear sum of individual rewards from all the agents and timesteps, which is a strict assumption and is abandoned in our work.

Plenty of work explores solving diverse RL problems with causality tools. From the aspect of the transfer ability of RL agents, [12, 6] learn factored representation and individual change factors for different domains in the stationary and non-stationary changes separately. More recently, [27] and [10] work in different ways to utilize the causal structure and variable dependencies to improve the generalizing capability of RL agents. For model-based RL, [46] discerns the confounders and [18] factorizes the state space in an identifiable way. In MARL, [8] discusses some open problems and [13] measures the causal influence of one agent on others as intrinsic rewards to motivate the agents to achieve higher change in the other agent's behavior. Another work related is LAIES [16], which also addresses the sparse reward in multi-agent learning and working along another line using the intrinsic rewards. However, they assume that the reward-relevant state components are known which is not practical in the real world. By contrast, our method can disentangle the reward-relevant state from the non-relevant component automatically.

B Background

Decentralized Partially Observable Markov Decision Process (Dec-POMDP). Dec-POMDP [23] is widely utilized in multi-agent reinforcement learning. It is defined by a tuple $\mathcal{M} = \langle N, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{O}, \Omega, \gamma \rangle$, where N represents the number of agents, $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively. At each timestep t, given an environment state (joint state) $\boldsymbol{s}_t$, each agent n observes observation $\boldsymbol{o}_t^n = \mathcal{O}(\boldsymbol{s}_t, n) : \mathcal{S} \times N \to \Omega$, and takes its individual action $\boldsymbol{a}_t^n$ and all the actions form a joint action $\boldsymbol{a}_t = [\boldsymbol{a}_t^1, \cdots, \boldsymbol{a}_t^N]$. Afterward, the agent n receives the team reward R_t based on the team reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. $\mathcal{P}(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t, \boldsymbol{a}_t)$ specifies the probability of transitioning to a new state $\boldsymbol{s}_{t+1}$ given the current state $\boldsymbol{s}_t$ and joint action $\boldsymbol{a}_t$. The objective for each agent is to find an optimal policy π^* that maximizes the discounted sum of team rewards, which is denoted as $\pi^* = \arg\max_\pi \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)]$, where γ represents the discount factor. The Dec-POMDP model is flexible and can be used in a wide range of multi-agent scenarios, making it a popular choice for coordination among multiple agents.

Episodic Reward Setting in Cooperative Games. Commonly, the team of agents receives a reward R_t immediately after the execution of the joint action $\boldsymbol{a}_t$ which consists of all the agents' action $\boldsymbol{a}_t^n$ at joint state $\boldsymbol{s}_t$. However, in the setting of episodic reinforcement learning, agents can only obtain one global reward feedback at the end of the trajectory. Let $\tau = (\boldsymbol{s}_1, \boldsymbol{a}_1, \boldsymbol{s}_2, \boldsymbol{a}_2, \cdots, \boldsymbol{s}_T, \boldsymbol{a}_T)$ denotes a trajectory of length T. Then, the team of agents can only observe an episodic reward R_{ep} at

timestep $t = T$, otherwise zero ($t \neq T$). Therefore, the goal of the team of agents is to maximize the trajectory return, $J = \mathbb{E}[\sum_{t=1}^{T}[\gamma^{t-1}R_t]] = \mathbb{E}[\gamma^{T-1}R_{\text{ep}}]$. In practice, it is common to assume that the episodic reward has some structure in nature, *i.e.*, reconstructing by an underlying reward function in a sum-decomposable form: $R_{\text{ep}} \approx \hat{R}_{\text{ep}} = \sum_{t=1}^{T}\gamma^{t-1}\mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)$, where γ is usually regarded to be 1.

C Limitations

The limitations of our work primarily stem from certain assumptions and constraints in the proposed method. Firstly, while our approach significantly enhances interpretability through causal-inspired spatial-temporal return decomposition, it relies on the assumption of a linear summation of team rewards over time, which may not always hold in more complex, real-world scenarios. Besides, although we allow for nonlinear mixtures of individual rewards in the spatial dimension, it requires the invertibility between the transformed reward and the individual reward. Future work could focus on extending our method to incorporate human knowledge to further relax the assumption of linear temporal and invertible spatial assumptions, making the proposed method more general.

D Broader Impact

The proposed framework has significant broader societal impacts in the field of multi-agent cooperative learning, as well as the real world. First, we provide a general approach to solving sparse rewards in Multi-agent Reinforcement Learning, especially enabling addressing the nonlinear team reward function, which is much more general and practical in the real world. Second, we enhance the transparency and credibility of algorithms through causal structure explanation, which can foster reliable and responsible decision-making in various fields, leading to better human-AI collaboration.

E Theoretical background

E.1 Causal Inference

A directed acyclic graph (DAG), $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$, can be deployed to represent a graphical criterion carrying out a set of conditions on the paths, where $\boldsymbol{V}$ and $\boldsymbol{E}$ denote the set of nodes and the set of directed edges, separately.

Definition E.1 (d-separation [26])**.** *A set of nodes $\boldsymbol{Z} \subseteq \boldsymbol{V}$ blocks the path p if and only if (1) p contains a chain $i \to m \to j$ or a fork $i \leftarrow m \to j$ such that the middle node m is in $\boldsymbol{Z}$, or (2) p contains a collider $i \to m \leftarrow j$ such that the middle node m is not in $\boldsymbol{Z}$ and such that no descendant of m is in $\boldsymbol{Z}$. Let $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$ be disjunct sets of nodes. If and only if the set $\boldsymbol{Z}$ blocks all paths from one node in $\boldsymbol{X}$ to one node in $\boldsymbol{Y}$, $\boldsymbol{Z}$ is considered to d-separate $\boldsymbol{X}$ from $\boldsymbol{Y}$, denoting as $(\boldsymbol{X} \perp_d \boldsymbol{Y} \mid \boldsymbol{Z})$.*

Definition E.2 (Global Markov Condition [35, 26])**.** *If, for any partition $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})$, $\boldsymbol{X}$ is d-separated from $\boldsymbol{Y}$ given $\boldsymbol{Z}$,* i.e.*, $\boldsymbol{X} \perp_d \boldsymbol{Y} \mid \boldsymbol{Z}$. Then the distribution P over $\boldsymbol{V}$ satisfies the global Markov condition on graph G, and can be factorized as, $P(\boldsymbol{X}, \boldsymbol{Y} \mid \boldsymbol{Z}) = P(\boldsymbol{X} \mid \boldsymbol{Z})P(\boldsymbol{Y} \mid \boldsymbol{Z})$. That is, $\boldsymbol{X}$ is conditionally independent of $\boldsymbol{Y}$ given $\boldsymbol{Z}$, writing as $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \mid \boldsymbol{Z}$.*

Definition E.3 (Faithfulness Assumption [35, 26])**.** *The variables, which are not entailed by the Markov Condition, are not independent of each other.*

Under the above assumptions, we can apply d-separation as a criterion to understand the conditional independencies from a given DAG $\mathcal{G}$. That is, for any disjoint subset of nodes $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} \subseteq \boldsymbol{V}$, $(\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \mid \boldsymbol{Z})$ and $\boldsymbol{X} \perp_d \boldsymbol{Y} \mid \boldsymbol{Z}$ are the necessary and sufficient condition of each other.

E.2 Exponential family

Definition E.4 (Exponential family)**.** *A univariate exponential family is a set of distributions whose probability density function can be written as*

$$p(x) = Q(x)Z(\theta)e^{T(x),\theta}, \tag{A1}$$

where $T : \mathbb{R} \to \mathbb{R}^k$ is called the sufficient statistic, $\theta \in \mathbb{R}^k$ is the natural parameter, $Q : \mathbb{R} \to \mathbb{R}$ the base measure and $Z(\theta)$ the normalization constant. The dimension $k \in \mathbb{N} \setminus \{0\}$ of the parameter is

always considered to be minimal, meaning that we can't rewrite the density p to have the form with a smaller $k' < k$. We call k the size of p.

Lemma E.5. *Consider an exponential family distribution with $k \geq 2$ components. If there exists $\alpha \in \mathbb{R}^k$ such that $T_k(x) = \sum_{i=1}^{k-1} \alpha_i T_i(x) + \alpha_k$, then $\alpha = 0$. In particular, the components of the sufficient statistic T are linearly independent.*

F Details of Theoretical Analysis

Below is the proof of Proposition 2.1 and Proposition 2.3. The proof is carried out through,

- identify transformed individual reward $\tilde{r}_t^n$, given the long-term return Q, joint state $\boldsymbol{s}_t$, joint action $\boldsymbol{a}_t$, and agent id n (Appendix F.1);
- identify individual reward vector $\boldsymbol{r}_t$, given the transformed individual reward $\tilde{r}_t^n$, joint state $\boldsymbol{s}_t$, joint action $\boldsymbol{a}_t$, and agent id n (Appendix F.2);
- distinguishes the individual rewards from the individual reward vector $\boldsymbol{r}_t$ (Appendix F.3);
- equivalence of policy learning with a monolithic transformation of the ground truth individual rewards (Appendix F.4).

For each part, we begin by clarifying the assumptions we made and then provide the mathematical proof.

F.1 Identifiability of transformed individual reward $\tilde{r}_t^n$

Assumption We assume that $\epsilon_{r,n,t}$ in Eq. 2 are i.i.d additive noise. From the weight-space view of Gaussian Process [43], equivalently, the causal models for transformed individual reward $\tilde{r}_t^n$, team reward R_t and long-term return Q can be represented as follows, respectively,

$$\begin{aligned} \tilde{r}_t^n &= \mathcal{R}_{f,g}(\boldsymbol{s}_t, \boldsymbol{a}_t, n) + \epsilon_{r,n,t} \\ &= W_{f,g}^T \phi_{f,g}(\boldsymbol{s}_t, \boldsymbol{a}_t) + \epsilon_{r,n,t}, \end{aligned} \tag{A2}$$

where $\phi_{f,g}$ denotes basis function sets.

Then we denote the variable set in the system by $\boldsymbol{V}$, with $\boldsymbol{V} = \{\boldsymbol{s}_{1,t}, \ldots, \boldsymbol{s}_{|\boldsymbol{s}|,t}, \boldsymbol{a}_{1,t}, \ldots, \boldsymbol{a}_{|\boldsymbol{a}|,t}, \tilde{r}_t^1, \cdots, \tilde{r}_t^N, R_t\}_{t=1}^T \cup Q$, and the variables form a Bayesian network $\mathcal{G}$. Note, we assume that there are possible edges only from $\boldsymbol{s}_{i,t} \in \boldsymbol{s}_t$ to $\tilde{r}_t^n$, from $\boldsymbol{a}_{j,t} \in \boldsymbol{a}_t$ to $\tilde{r}_t^n$, from $\tilde{r}_t^n$ to R_t, and from R_t to Q in $\mathcal{G}$.

Following the above assumption, we first rewrite the function to calculate trajectory-wise long-term return Q in Eq. 2 as,

$$\begin{aligned} Q &= \sum_{t=1}^{T} R_t = \sum_{t=1}^{T} \sum_{n=1}^{N} \tilde{r}_t^n \\ &= \sum_{t=1}^{T} \left(\sum_{n=1}^{N} \left(\mathcal{R}_{f,g}(\boldsymbol{s}_t, \boldsymbol{a}_t, n) + \epsilon_{r,n,t} \right) \right) \\ &= \sum_{t=1}^{T} \sum_{n=1}^{N} W_{f,g}^T \phi_{f,g}(\boldsymbol{s}_t, \boldsymbol{a}_t) + \sum_{t=1}^{T} \sum_{n=1}^{N} \epsilon_{r,n,t} \\ &= W_{f,g}^T \sum_{t=1}^{T} \sum_{n=1}^{N} \phi_{f,g}(\boldsymbol{s}_t, \boldsymbol{a}_t) + \sum_{t=1}^{T} \sum_{n=1}^{N} \epsilon_{r,n,t}, \end{aligned} \tag{A3}$$

For simplicity, we replace the components in Eq. A3 by,

$$\begin{aligned} \zeta(X) &= \sum_{t=1}^{T} \sum_{n=1}^{N} \phi_{f,g}(\boldsymbol{s}_t, \boldsymbol{a}_t, n), \\ E &= \sum_{t=1}^{T} \sum_{n=1}^{N} \epsilon_{r,n,t}, \end{aligned} \tag{A4}$$

where $X := [\boldsymbol{s}_t, \boldsymbol{a}_t, n]_{n=1,t=1}^{N,T}$ representing the concatenation of the covariates $\boldsymbol{s}_t$, $\boldsymbol{a}_t$, n from $n = 1$ to N and from $t = 1$ to T. Consequently, we derive the following equation,

$$Q = W^T \zeta(X) + E. \tag{A5}$$

Then we can obtain a closed-form solution of W^T in Eq. A5 by modeling the dependencies between the covariates X_τ and response variables Q_τ, where both are continuous. One classical approach to finding such a solution involves minimizing the quadratic cost and incorporating a weight-decay regularizer to prevent overfitting. Specifically, we define the cost function as,

$$C(W) = \frac{1}{2} \sum_{X_\tau, Q_\tau \sim \mathcal{D}} (Q_\tau - W^T \zeta(X_\tau))^2 + \frac{1}{2} \lambda \|W\|^2, \tag{A6}$$

where τ represents trajectories consisting of state-action-id combinations X_τ and long-term returns Q_τ, which are sampled from the replay buffer $\mathcal{D}$. λ is the weight-decay regularization parameter. To find the closed-form solution, we differentiate the cost function with respect to W and set the derivative to zero:

$$\frac{\partial C(W)}{\partial W} = 0. \tag{A7}$$

Solving this equation will yield the closed-form solution for W^T, *i.e.*,

$$W \quad = (\lambda I_d + \zeta \zeta^T)^{-1} \zeta Q = \zeta (\zeta^T \zeta + \lambda I_n)^{-1} Q, \tag{A8}$$

where I_n denotes the identity matrix with size n. Therefore, W, which indicates the causal structure and strength of the edge, can be identified from the observed data. In summary, given trajectory-wise long-term return Q, the causal structure for the generation of the transformed individual rewards $\tilde{r}_t$ and team reward R_t are identifiable.

F.2 Identifiability of individual reward vector $\boldsymbol{r}_t$

Now we solve the problem of identifying the individual reward vector $\boldsymbol{r}_t$, given the joint state $\boldsymbol{s}_t$, joint action $\boldsymbol{a}_t$, transformed individual rewards. We assume that there is no direct causal edge within individual rewards r_t^n. The following proof is the application of the theorems in iVAE [14], and we further go beyond permutation-invariant latent by leveraging the agent's action.

Assumptions Let $\mathcal{X} \subset \mathbb{R}^{D \times N}$ denotes the combination of joint state, joint action, and agent id for simplicity, where D is the dimension of $\mathcal{X}^n$ the combination of joint state, joint action and a single id for the n-th agent. The agent number is N. Let $\mathcal{R} \subset \mathbb{R}^N$ be the individual rewards for N agents organized by the order of the agent's action in the joint action. Let $\tilde{\mathcal{R}} \subset \mathbb{R}^N$ be the transformed individual rewards for N agents, also organized by the order of the agents' actions in the joint action. Therefore, we want to identify $\mathcal{R}$, given $\mathcal{X}$ and $\tilde{\mathcal{R}}$. They have the following causal relationships,

$$\mathcal{X} \to \mathcal{R} \to \tilde{\mathcal{R}}. \tag{A9}$$

We suppose that $\mathcal{X}$, $\mathcal{R}$, $\tilde{\mathcal{R}}$ are open sets.

Therefore, we have the following conditional generative model for the generation of the involved data,

$$p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{r}}, \boldsymbol{r} \mid \boldsymbol{x}) = p_g(\tilde{\boldsymbol{r}} \mid \boldsymbol{r}) p_{\boldsymbol{T}, \boldsymbol{\lambda}}(\boldsymbol{r} \mid \boldsymbol{x}) = p_g(\tilde{\boldsymbol{r}} \mid \boldsymbol{r}) \prod_{n=1}^{N} p_{\boldsymbol{T}^n, \boldsymbol{\lambda}^n}(\boldsymbol{r}^n \mid \boldsymbol{x}^n), \tag{A10}$$

where we define,

$$p_g(\tilde{\boldsymbol{r}} \mid \boldsymbol{r}) = p_\epsilon(\tilde{\boldsymbol{r}} - g(\boldsymbol{r})), \tag{A11}$$

which means that the value of $\tilde{\boldsymbol{r}}$ can be decomposed as $\tilde{\boldsymbol{r}} = g(\boldsymbol{r}) + \epsilon$ where ϵ is an independent noise variable with probability density function p_ϵ, *i.e.*, ϵ is independent of $\boldsymbol{r}$ or g. Therefore, the set $\{\tilde{\boldsymbol{r}} \in \tilde{\mathcal{R}} \mid \phi_\epsilon = 0\}$ has measure zero, where ϕ_ϵ is the characteristic function of the density p_ϵ.

We use $\Theta = \{\boldsymbol{\theta} := (g, \boldsymbol{T}, \boldsymbol{\lambda})\}$ to denote the involved parameters. Recall our assumption that, g is the invertible function, and the $\tilde{\mathcal{R}}$ is generated through the invertible function g from $\mathcal{R}$. We denote by g^{-1} the inverse defined from $\tilde{\mathcal{R}} \to \mathcal{R}$. Note that, since the g is invertible, it is also bijective (*i.e.*, both injective and surjective). Also, the g has all second order cross derivatives We denote by $\boldsymbol{T}(\boldsymbol{r}) := (\boldsymbol{T}^1(r^1), \cdots, \boldsymbol{T}^N(r^N)) = (T_1^1(r^1), \cdots, T_K^1(r^1), \cdots, T_1^N(r^N), \cdots, T_K^N(r^N)) \in \mathbb{R}^{N,K}$ the vector of sufficient statistics of the probability density function, and $\boldsymbol{\lambda}(\boldsymbol{x}) = (\boldsymbol{\lambda}^1(\boldsymbol{x}^1), \cdots, \boldsymbol{\lambda}^N(\boldsymbol{x}^N)) =$

$(\lambda_1^1(\boldsymbol{x}^1), \cdots, \lambda_K^1(\boldsymbol{x}^N), \cdots, \lambda_1^N(\boldsymbol{x}^N) \cdots, \lambda_K^N(\boldsymbol{x}^N)) \in \mathbb{R}^{N,K}$ the corresponding parameters, crucially depending on $\boldsymbol{x}$ for

$$p_{\boldsymbol{T},\boldsymbol{\lambda}}(\boldsymbol{r} \mid \boldsymbol{x}) = \prod_n \frac{Q^n(r^n)}{Z^n(\boldsymbol{x}^n)} \exp\left[\sum_{k=1}^K T_k^n(r^n)\lambda_k^n(\boldsymbol{x}^n)\right], \tag{A12}$$

where Q^n is the base measure, $Z^n(\boldsymbol{x}^n)$ is the normalizing constant, and the dimension of each sufficient statistic, K. Such exponential families have universal approximation capabilities [36].

Following iVAE [14], we further have the following assumptions.

Assumption F.1. *The sufficient statistics $\boldsymbol{T}_k^n$ in Eq. A12 are twice differentiable, and $(\boldsymbol{T}_k^n)_{1\leq k\leq K, 1\leq n\leq N}$ are linearly independent on any subset of $\tilde{\mathcal{R}}$ of measure greater than zero.*

Assumption F.2. *There exist $NK+1$ distinct point $\boldsymbol{x}^0, \cdots, \boldsymbol{x}^{NK}$ such that the matrix*

$$L = (\boldsymbol{\lambda}(\boldsymbol{x}^1) - \boldsymbol{\lambda}(\boldsymbol{x}^0), \cdots, \boldsymbol{\lambda}(\boldsymbol{x}^{NK}) - \boldsymbol{\lambda}(\boldsymbol{x}^0)), \tag{A13}$$

of size $NK \times NK$ is invertible.

Then we demonstrate that the vector $\boldsymbol{r}$, *i.e.*, the goal of credit assignment are identifiable up to a class of transformation.

Recall $\Theta = \{\boldsymbol{\theta} := (g, \boldsymbol{T}, \boldsymbol{\lambda}\})$ to denote the involved parameters. We give the following definitions:

Definition F.3. *Let $\sim$ be an equivalence relation on Θ. We say that,*

$$p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{r}}, \boldsymbol{r}) = p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{r}} \mid \boldsymbol{r}) p_{\boldsymbol{\theta}}(\boldsymbol{r}), \tag{A14}$$

is identifiable up to $\sim$ (or $\sim$-identifiable) if

$$p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{r}}) = p_{\tilde{\boldsymbol{\theta}}}(\tilde{\boldsymbol{r}}) \Rightarrow \tilde{\boldsymbol{\theta}} \sim \boldsymbol{\theta}, \tag{A15}$$

where $(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) \in \Theta$. The elements of the quotient space $\Theta/\sim$ are called the identifiability classes.

Definition F.4. *There are two equivalence relations on the set of parameters Θ. Let $\sim$ be the equivalence relation on Θ defined as follows:*

$$(g, \boldsymbol{T}, \boldsymbol{\lambda}) \sim (\tilde{g}, \tilde{\boldsymbol{T}}, \tilde{\boldsymbol{\lambda}}) \Leftrightarrow \exists \boldsymbol{A}, \boldsymbol{c} \mid \boldsymbol{T}(g^{-1}(\tilde{\boldsymbol{r}})) = \boldsymbol{A}\tilde{\boldsymbol{T}}(\tilde{g}^{-1}(\tilde{\boldsymbol{r}}) + \boldsymbol{c}, \forall \tilde{\boldsymbol{r}} \in \tilde{\mathcal{R}}, \tag{A16}$$

where $\boldsymbol{A}$ is an $NK \times NK$ matrix and $\boldsymbol{c}$ is a vector. If $\boldsymbol{A}$ is invertible, we denote this relation by $\sim_A$. If $\boldsymbol{A}$ is a block permutation matrix, we denote it by $\sim_P$.

We begin with proving that the parameters $(g, \boldsymbol{T}, \boldsymbol{\lambda})$ are $\sim_A$-identifiable, which means that 1) if there is no noise, we can learn the transformation g transforms the $\tilde{\boldsymbol{r}}$ into individual rewards $\boldsymbol{r} = g^{-1}(\tilde{\boldsymbol{r}})$ that are equal to the ground truth individual rewards, up to a linear invertible transformation (the matrix $\boldsymbol{A}$) and point-wise nonlinearities (in the form of $\boldsymbol{T}$ and $\tilde{\boldsymbol{T}}$). 2) if with noise, we obtain the posteriors of the individual rewards $\boldsymbol{r}$ up to an analogous indeterminacy. We prove the identifiability result in several steps.

- In the first step, we demonstrate that, given the assumption that,

  $$\{\tilde{\boldsymbol{r}} \in \tilde{\mathcal{R}} \mid \phi_\epsilon = 0\}, \tag{A17}$$

  has measured zero, it is possible to use a simple convolutional trick to transform the equality of observed data distributions into the equality of noiseless distributions. In other words, it simplifies the noisy case into a noiseless case.
- The second step consists of removing all terms that are either a function of transformed individual rewards $\tilde{\boldsymbol{r}}$ or observed $\boldsymbol{x}$. This is done by introducing the points provided by Eq. A13 and using $\boldsymbol{x}_0$ as a "pivot". This is simply done in equations.
- The last step of the proof is to show that the linear transformation is invertible, thus resulting in an equivalence relation.

Step I We introduce here the volume of a matrix denoted $\text{vol}(A)$ as the product of the singular values of A. When A is full column rank, $\text{vol}(A) = \sqrt{\det A^T A}$, and when A is invertible, $\text{vol}(A) = |\det A|$. The matrix volume can be used in the change of variable formula as a replacement for the absolute determinant of the [2]. This is most useful when the Jacobian is a rectangular matrix ($N < D$). Suppose we have two sets of parameters $(g, \boldsymbol{T}, \boldsymbol{\lambda})$ and $(\tilde{g}, \tilde{\boldsymbol{T}}, \tilde{\boldsymbol{\lambda}})$ such that $p_{g,\boldsymbol{T},\boldsymbol{\lambda}}(\tilde{\boldsymbol{r}}|\boldsymbol{x}) = p_{\tilde{g},\tilde{\boldsymbol{T}},\tilde{\boldsymbol{\lambda}}}(\tilde{\boldsymbol{r}} \mid \boldsymbol{x})$ for all pairs $(\tilde{\boldsymbol{r}}, \boldsymbol{x})$. Then:

$$\begin{aligned}
\int_{\mathcal{R}} \left[\prod_n p_{\boldsymbol{T},\boldsymbol{\lambda}}(\boldsymbol{r}^n|\boldsymbol{x}^n)\right] p_g(\tilde{\boldsymbol{r}}|\boldsymbol{r})d\boldsymbol{r} &= \int_{\mathcal{R}} \left[p_{\boldsymbol{T},\boldsymbol{\lambda}}(\boldsymbol{r}|\boldsymbol{x})\right] p_g(\tilde{\boldsymbol{r}}|\boldsymbol{r})d\boldsymbol{r} \\
\Rightarrow \quad \int_{\mathcal{R}} \left[p_{\boldsymbol{T},\boldsymbol{\lambda}}(\boldsymbol{r}|\boldsymbol{x})\right] p_g(\tilde{\boldsymbol{r}}|\boldsymbol{r})d\boldsymbol{r} &= \int_{\mathcal{R}} \left[p_{\tilde{\boldsymbol{T}},\tilde{\boldsymbol{\lambda}}}(\boldsymbol{r} \mid \boldsymbol{x})\right] p_{\tilde{g}}(\tilde{\boldsymbol{r}}|\boldsymbol{r})d\boldsymbol{r} \\
\Rightarrow \quad \int_{\mathcal{R}} \left[p_{\boldsymbol{T},\boldsymbol{\lambda}}(\boldsymbol{r}|\boldsymbol{x})\right] p_\epsilon(\tilde{\boldsymbol{r}} - g(\boldsymbol{r}))d\boldsymbol{r} &= \int_{\mathcal{R}} \left[p_{\tilde{\boldsymbol{T}},\tilde{\boldsymbol{\lambda}}}(\boldsymbol{r} \mid \boldsymbol{x})\right] p_\epsilon(\tilde{\boldsymbol{r}} - \tilde{g}(\boldsymbol{r}))d\boldsymbol{r} \\
\Rightarrow \quad \int_{\tilde{\mathcal{R}}} \left[p_{\boldsymbol{T},\boldsymbol{\lambda}}(g^{-1}(\bar{\tilde{\boldsymbol{r}}})|\boldsymbol{x})\text{vol}(J)_{g^{-1}(\bar{\tilde{\boldsymbol{r}}})}\right] p_\epsilon(\tilde{\boldsymbol{r}} - \bar{\tilde{\boldsymbol{r}}})d\bar{\tilde{\boldsymbol{r}}} &= \int_{\tilde{\mathcal{R}}} \left[p_{\tilde{\boldsymbol{T}},\tilde{\boldsymbol{\lambda}}}(\tilde{g}^{-1}(\bar{\tilde{\boldsymbol{r}}})|\boldsymbol{x})\text{vol}(J)_{\tilde{g}^{-1}(\bar{\tilde{\boldsymbol{r}}})}\right] p_\epsilon(\tilde{\boldsymbol{r}} - \bar{\tilde{\boldsymbol{r}}})d\bar{\tilde{\boldsymbol{r}}},
\end{aligned} \tag{A18}$$

where J denotes the Jacobian, and we made the change of variable $\bar{\tilde{\boldsymbol{r}}} = g(\boldsymbol{r})$ and $\bar{\tilde{\boldsymbol{r}}} = \tilde{g}(\boldsymbol{r})$. Then, we use the following to replace the terms in the Equation,

$$\tilde{p}_{\boldsymbol{T},\boldsymbol{\lambda},g,\boldsymbol{x}}(\tilde{\boldsymbol{r}}) = p_{\boldsymbol{T},\boldsymbol{\lambda}}(g^{-1}(\tilde{\boldsymbol{r}})|\boldsymbol{x}^n)\text{vol}(J)_{g^{-1}}(\tilde{\boldsymbol{r}})\mathbf{1}_{\tilde{\mathcal{R}}}(\tilde{\boldsymbol{r}}) = p_{\boldsymbol{T},\boldsymbol{\lambda}}(g^{-1}(\tilde{\boldsymbol{r}})|\boldsymbol{x})\text{vol}(J)_{g^{-1}}(\tilde{\boldsymbol{r}})\mathbf{1}_{\tilde{\mathcal{R}}}(\tilde{\boldsymbol{r}}), \tag{A19}$$

and get the following,

$$\begin{aligned}
\Rightarrow \quad & \int_{\mathbb{R}^N} \tilde{p}_{\boldsymbol{T},\boldsymbol{\lambda},g,\boldsymbol{x}}(\bar{\tilde{\boldsymbol{r}}})p_e(\tilde{\boldsymbol{r}} - \bar{\tilde{\boldsymbol{r}}})d\bar{\tilde{\boldsymbol{r}}} = \int_{\mathbb{R}^N} \tilde{p}_{\tilde{\boldsymbol{T}},\tilde{\boldsymbol{\lambda}},\tilde{g},\boldsymbol{x}}(\bar{\tilde{\boldsymbol{r}}})p_e(\tilde{\boldsymbol{r}} - \bar{\tilde{\boldsymbol{r}}})d\bar{\tilde{\boldsymbol{r}}} \\
\Rightarrow \quad & (\tilde{p}_{\boldsymbol{T},\boldsymbol{\lambda},g,\boldsymbol{x}} * p_\epsilon)(\tilde{\boldsymbol{r}}) = (p_{\tilde{\boldsymbol{T}},\tilde{\boldsymbol{\lambda}},\tilde{g},\boldsymbol{x}} * p_\epsilon)(\tilde{\boldsymbol{r}}),
\end{aligned} \tag{A20}$$

where $*$ denotes the convolution operator. Then we used $F[\cdot]$ to designate the Fourier transform, and where $\phi_\epsilon = F[p_\epsilon]$ (by definition of the characteristic function).

$$\begin{aligned}
\Rightarrow \quad & F[\tilde{p}_{\boldsymbol{T},\boldsymbol{\lambda},g,\boldsymbol{x}}](\omega)p_\epsilon(\omega) = F[\tilde{p}_{\tilde{\boldsymbol{T}},\tilde{\boldsymbol{\lambda}},\tilde{g},\boldsymbol{x}}](\omega)p_\epsilon(\omega) \\
\Rightarrow \quad & F[\tilde{p}_{\boldsymbol{T},\boldsymbol{\lambda},g,\boldsymbol{x}}](\omega) = F[\tilde{p}_{\tilde{\boldsymbol{T}},\tilde{\boldsymbol{\lambda}},\tilde{g},\boldsymbol{x}}](\omega),
\end{aligned} \tag{A21}$$

$\phi_e(\omega)$ from both sides are dropped as it is non-zero almost everywhere $\{\tilde{\boldsymbol{r}} \in \tilde{\mathcal{R}} \mid \phi_\epsilon = 0\}$. Then we get

$$\tilde{p}_{\boldsymbol{T},\boldsymbol{\lambda},g,\boldsymbol{x}}(\tilde{\boldsymbol{r}}) = \tilde{p}_{\tilde{\boldsymbol{T}},\tilde{\boldsymbol{\lambda}},\tilde{g},\boldsymbol{x}}(\tilde{\boldsymbol{r}}), \tag{A22}$$

Eq. A22 is valid for all $(\tilde{\boldsymbol{r}}, \boldsymbol{x}) \in \tilde{\mathcal{R}} \times \mathcal{X}$. It basically says that for the distributions to be the same after adding the noise, the noise-free distributions have to be the same. Note that $\tilde{\boldsymbol{r}}$ is a general variable and we are actually dealing with the noise-free probability densities.

Step II By taking the logarithm on both sides of Eq. A22 and replacing $p_{\boldsymbol{T},\boldsymbol{\lambda}}$ by its expression from Eq. A10, we get:

$$\begin{aligned}
\log\left[\tilde{p}_{\boldsymbol{T},\boldsymbol{\lambda},g,\boldsymbol{x}}(\tilde{\boldsymbol{r}})\right] &= \log\left[p_{\boldsymbol{T},\boldsymbol{\lambda}}(g^{-1}(\tilde{\boldsymbol{r}})|\boldsymbol{x})\text{vol}(J)_{g^{-1}}(\tilde{\boldsymbol{r}})\mathbf{1}_{\tilde{\mathcal{R}}}(\tilde{\boldsymbol{r}})\right] \\
&= \log\left[\text{vol}(J)_{g^{-1}}(\tilde{\boldsymbol{r}}) + p_{\boldsymbol{T},\boldsymbol{\lambda}}(g^{-1}(\tilde{\boldsymbol{r}}) \mid \boldsymbol{x}^n)\right] \\
&= \log\left[\text{vol}(J)_{g^{-1}}(\tilde{\boldsymbol{r}})\right] + \log\left[\prod_n \frac{Q^n([g^{-1}(\tilde{\boldsymbol{r}})]^n)}{Z^n(\boldsymbol{x}^n)} \exp\left[\sum_{k=1}^K T_k^n([g^{-1}(\tilde{\boldsymbol{r}})]^n)\lambda_k^n(\boldsymbol{x}^n)\right]\right] \\
&= \log\left[\text{vol}(J)_{g^{-1}}(\tilde{\boldsymbol{r}})\right] + \sum_{n=1}^N \left[\log Q^n(g^{-1,n}(\tilde{\boldsymbol{r}})) - \log Z^n(\boldsymbol{x}^n) + \sum_{k=1}^K T_k^n(g^{-1,n}(\tilde{\boldsymbol{r}}))\lambda_k^n(\boldsymbol{x}^n)\right],
\end{aligned} \tag{A23}$$

where we denote the n-th element of $g^{-1}(\tilde{\boldsymbol{r}})$ by $[g^{-1}(\tilde{\boldsymbol{r}})]^n$. Therefore,

$$\log\left[\text{vol}(J)_{g^{-1}}(\tilde{\boldsymbol{r}})\right] + \sum_{n=1}^{N}\left(\log Q^n([g^{-1}(\tilde{\boldsymbol{r}})]^n) - \log Z^n(\boldsymbol{x}^n) + \sum_{k=1}^{K} T_k^n[(g^{-1}(\tilde{\boldsymbol{r}})]^n)\lambda_k^n(\boldsymbol{x}^n)\right) =$$
$$\log\left[\text{vol}(J)_{\tilde{g}^{-1}}(\tilde{\boldsymbol{r}})\right] + \sum_{n=1}^{N}\left(\log \tilde{Q}^n([\tilde{g}^{-1}(\tilde{\boldsymbol{r}})]^n) - \log \tilde{Z}^n(\boldsymbol{x}^n) + \sum_{k=1}^{K} \tilde{T}_k^n([\tilde{g}^{-1}(\tilde{\boldsymbol{r}})]^n)\tilde{\lambda}_k^n(\boldsymbol{x}^n)\right). \tag{A24}$$

Let $\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{NK}$ be the points provided by Assumption F.2 and define $\lambda(\boldsymbol{x}) = \lambda(\boldsymbol{x}) - \lambda(\boldsymbol{x}^0)$. We plug each of those $\boldsymbol{x}^i$ in Eq. A24 to obtain $NK+1$ such equations. We subtract the first equation for $\boldsymbol{x}^0$ from the remaining NK equations to get for $l = 1, ..., NK$:

$$\langle \boldsymbol{T}(g^{-1}(\tilde{\boldsymbol{r}})), \bar{\lambda}(\boldsymbol{x}^l)\rangle + \sum_n \log\frac{Z^n(\boldsymbol{x}^0)}{Z^n(\boldsymbol{x}^l)} = \langle \tilde{\boldsymbol{T}}(\tilde{g}^{-1}(\tilde{\boldsymbol{r}})), \tilde{\lambda}(\boldsymbol{x})\rangle + \sum_n \log\frac{\tilde{Z}^n(\boldsymbol{x}^0)}{\tilde{Z}^n(\boldsymbol{x}^l)}. \tag{A25}$$

Let L be the matrix defined in Assumption F.2, and $\tilde{L}$ similarly defined for $\tilde{\lambda}$ (L is not necessarily invertible). Define $b^l = \sum_n \log\frac{\tilde{Z}^n(\boldsymbol{x}^0)Z^n(\boldsymbol{x}^l)}{Z^n(\boldsymbol{x}^0)\tilde{Z}^n(\boldsymbol{x}^l)}$ and b the vector of all b^l for $l = 1, ..., NK$. Expressing Eq. A25 for all points $\boldsymbol{x}^l$ in matrix form, we get:

$$L^T\boldsymbol{T}(g^{-1}(\tilde{\boldsymbol{r}})) = \tilde{L}^T\tilde{\boldsymbol{T}}(\tilde{g}^{-1}(\tilde{\boldsymbol{r}})) + \boldsymbol{b}. \tag{A26}$$

Then, after multiplying both sides of Eq. A26 by the transpose of the inverse of L^T from the left, we obtain:

$$\boldsymbol{T}(g^{-1}(\tilde{\boldsymbol{r}})) = [L^T]^{-1}\tilde{L}^T\tilde{\boldsymbol{T}}(\tilde{g}^{-1}(\tilde{\boldsymbol{r}})) + [L^T]^{-1}\boldsymbol{b}. \tag{A27}$$

Denoting $A = [L^T]^{-1}\tilde{L}$ and $\boldsymbol{c} = [L^T]^{-1}\boldsymbol{b}$, we have

$$\boldsymbol{T}(g^{-1}(\tilde{\boldsymbol{r}})) = A\tilde{\boldsymbol{T}}(\tilde{g}^{-1}(\tilde{\boldsymbol{r}})) + \boldsymbol{c}. \tag{A28}$$

Step III Now by definition of $\boldsymbol{T}$ and according to Assumption F.1, its Jacobian exists and is an $NK \times N$ matrix of rank N, which implies that the Jacobian of $\tilde{\boldsymbol{T}} \circ g^{-1}$ exists and is of rank N and so is A.

We distinguish two cases in the following.

If $K = 1$, then this means that A is invertible (because A is $N \times N$).

If $K > 1$, define $\bar{\boldsymbol{r}} = g^{-1}(\tilde{\boldsymbol{r}})$ and $\boldsymbol{T}^n(\tilde{\boldsymbol{r}}^n) = (T_1^n(\bar{\boldsymbol{r}}), \ldots, T_K^N(\bar{\boldsymbol{r}}))$.

Lemma F.5. *Consider a strongly exponential distribution of size $k \geq 2$ with sufficient statistic $\boldsymbol{T}(x) = (T^1(x), \cdots, T^K(x))$. Further, assume that $\boldsymbol{T}$ is differentiable almost everywhere. Then there exists K distinct values x_1 to x_K such that $(\boldsymbol{T}'(x_1), \cdots, \boldsymbol{T}'(x_k))$ are linearly independent in $\mathbb{R}^K$.*

According to the Lamma, for each $n \in \{1, \ldots, N\}$ there exist K points $(\bar{\boldsymbol{r}}_1^n, \ldots, \bar{\boldsymbol{r}}_K^n)$ such that $(\boldsymbol{T}^{n\prime}(\bar{\boldsymbol{r}}_1^n), \ldots, \boldsymbol{T}^{n\prime}(\bar{\boldsymbol{r}}_K^n))$ are linearly independent. Collect those points into K vectors $(\bar{\boldsymbol{r}}_1 1, \ldots, \bar{\boldsymbol{r}}_K)$, and concatenate the K Jacobians $J_{\boldsymbol{T}}(\bar{\boldsymbol{r}}_l)$ evaluated at each of those vectors horizontally into the matrix $Q = (J_{\boldsymbol{T}}(\bar{\boldsymbol{r}}_1), \ldots, J_{\boldsymbol{T}}(\bar{\boldsymbol{r}}^K))$ (and similarly define $\tilde{Q}$ as the concatenation of the Jacobians of $\tilde{\boldsymbol{T}}(\tilde{g}^{-1} \circ g(\bar{\boldsymbol{r}}))$ evaluated at those points). Then the matrix Q is invertible (through a combination of Lemma F.5 and the fact that each component of $\tilde{T}$ is univariate). By differentiating Eq. A28 for each $\tilde{\boldsymbol{r}}_l$, we get (in matrix form):

$$Q = A\tilde{Q}. \tag{A29}$$

The invertibility of Q implies the invertibility of A and $\tilde{Q}$. Hence, Eq. A28 and the invertibility of A mean that $(\tilde{g}, \tilde{\boldsymbol{T}}, \tilde{\boldsymbol{\lambda}}) \sim (g, \boldsymbol{T}, \boldsymbol{\lambda})$. Moreover, we have the following observations:

- the invertibility of A and L imply that $\tilde{L}$ is invertible,
- because the Jacobian of $\tilde{\boldsymbol{T}} \circ \tilde{g}^{-1}$ is full rank and $\tilde{g}$ is injective (hence its Jacobian is full rank too), $J_{\tilde{\boldsymbol{T}}}$ has to be full rank too, and $\tilde{\boldsymbol{T}}_k^{n\prime}(\boldsymbol{r}) \neq 0$ almost everywhere.
- the real equivalence class of identifiability may actually be narrower than what is defined by $\sim$, as the matrix A and the vector $\boldsymbol{c}$ here have very specific forms, and are functions of $\boldsymbol{\lambda}$ and $\tilde{\boldsymbol{\lambda}}$.

Step IV Following iVAE [14], the assumption mentioned still holds, we then remove the linear indeterminacy A, and reduce the equivalence relation to $\sim_P$.

First, we want to show the case of $K \geq 2$. Recall we observe the following from the last part of the proof:

$$\boldsymbol{T}(g^{-1}(\tilde{\boldsymbol{r}})) = A\tilde{\boldsymbol{T}}(\tilde{g}^{-1}(\tilde{\boldsymbol{r}})) + \boldsymbol{c}. \tag{A30}$$

for an invertible $A \in \mathbb{R}^{NK \times NK}$. Following iVAE [14], we index A by (i, l, a, b) where $1 \leq i \leq N, 1 \leq l \leq K$ and $1 \leq a \leq N, 1 \leq b \leq K$ to denote the rows and columns separately. Define $\boldsymbol{v}(\boldsymbol{r}) = \tilde{g}^{-1} \circ g(\boldsymbol{r}) : \mathcal{R} \to \mathcal{R}$ where $\boldsymbol{v}$ is bijective since both g and g^{-1} are injective. Then we show $v_i(\boldsymbol{r})$ is a function of only one r_{j_i}, for all i. Define $v_i^s := \frac{\partial v_1}{\partial r_s}(\boldsymbol{r})$ and $v_i^{st} := \frac{\partial^2 v_i}{\partial r_s \partial r_t}(\boldsymbol{r})$. For each $1 \leq i \leq N, 1 \leq l \leq K$, we obtain

$$\delta_{is} T_l^{i\prime}(r^i) = \sum_{a,b} A_{i,l,a,b} \tilde{T}_b^{a\prime}(v_a(\boldsymbol{r})) v_a^s(\boldsymbol{r}), \tag{A31}$$

by differentiating Eq. A30 with respect to r_s and by differentiating Eq. A31 with respect to $r_q, q > s$, we have,

$$0 = \sum_{a,b} A_{i,l,a,b} \left(\tilde{T}_b^{a\prime}(v_a(\boldsymbol{r})) v_a^{s,q}(\boldsymbol{r}) + \tilde{T}_b^{a,\prime\prime}(v_a(\boldsymbol{r})) v_a^s(\boldsymbol{r}) v_a^q(\boldsymbol{z}) \right), \tag{A32}$$

which is valid for all pairs $(s, q), q > s$.

According to iVAE, the Jacobian of $\boldsymbol{v}$ at each $\boldsymbol{r}$ has at most one non-zero entry in each row. Since $J_{\boldsymbol{v}}$ is invertible and continuous, the locations of the non-zero entries are fixed and do not change as a function of $\boldsymbol{r}$. Therefore, we know that, the $\tilde{g}^{-1} \circ g$ is point-wise nonlinearity.

Then we combine the mentioned observation of $\tilde{g} \circ g^{-1}$ with the results of Eq. A28 from the last part of the proof. Let $\bar{\boldsymbol{T}}(\boldsymbol{r} = \boldsymbol{v}(\boldsymbol{r}) \tilde{+} A^{-1}\boldsymbol{c}$. $\bar{\boldsymbol{T}}$ is a composition of a permutation and pointwise nonlinearity. Without any loss of generality, the permutation in $\bar{\boldsymbol{T}}$ is assumed to be the identity. Then we have,

$$\boldsymbol{T}(\boldsymbol{r}) = A\bar{\boldsymbol{T}}(r). \tag{A33}$$

Let $D = A^{-1}$, Eq. A33 is valid for every component in,

$$\bar{\boldsymbol{T}}_l^i(\boldsymbol{r}^i) = \sum_{a,b} D_{i,l,a,b} \boldsymbol{T}_b^a(r_a). \tag{A34}$$

Through differentiating with respect to r_s where $s \neq i$, we have,

$$0 = \sum_b D_{i,l,s,b} \boldsymbol{T}_b^{s,\prime}(\boldsymbol{r}_s), \tag{A35}$$

which is valid for all l and all $s \neq i$. By the Lemma 1 in iVAE [14], we get $D_{i,l,a,b} = 0$ for all $1 \leq b \leq K$. Then we know that matrix D has a block diagonal form:

$$D = \begin{pmatrix} D_1 & & \\ & \cdots & \\ & & D_n \end{pmatrix}. \tag{A36}$$

Therefore, A has the same black diagonal form. Each block i transforms $\boldsymbol{T}^i(\boldsymbol{r})$ into $\bar{\boldsymbol{T}}^i(\boldsymbol{r})$, which demonstrate A is necessarily a permutation matrix.

According to iVAE, in the case of $K = 1$, $\tilde{g} \circ g$ is also a point-wise nonlinearity.

Therefore, we can identify the individual reward vector $\boldsymbol{r}$ up to its invertible transformation.

F.3 Identify r_t^n from r for each agents

Through the iVAE's application in our setting, we already prove that, we can recover a latent vector $\boldsymbol{r}_t$, as well as the behind parameters, which also includes the causal structures. However, in the setting of iVAE, it is only required to identify a permutation-invariant latent vector. So now the key problem is how we determine which element of the latent vector $\boldsymbol{r}$ should be responsible for the agent n.

We assume that, the individual rewards are the outcome of the joint state and the agent's own action, therefore, there is always a causal edge from the agent n's individual action into its own individual reward r_t^n, which can help us extend the conclusion of identifying the permutation-invariant latent vector into identifying a permutation-variant and agent-aware latent vector.

F.4 Equivalence of Policy Learning with Nonlinear Invertible Transformations of Individual Rewards

Here is the proof of Proposition 2.3. Assume $k(\cdot) : \mathbb{R} \to \mathbb{R}$ is a monotonically increasing transformation in Proposition 2.3. We now prove that, it is equivalent to optimizing the policy π by the guidance of individual reward r_t and the transformation $k(r_t)$. For simplicity, we ignore the agent index n, *i.e.*, $r_t = r_t^n$.

Assume that optimal policy π^*:

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_{\boldsymbol{a}_t \sim \pi} \sum_{t=1}^{T} \gamma^{t-1} r_t \\ \pi^* &= \arg\max_{\pi \in \Pi} V^\pi(s), \end{aligned} \tag{A37}$$

$$\mathbb{E}_{\boldsymbol{a}_t \sim \pi^*} \sum_{t=1}^{T} \gamma^{t-1} r_t \geq \mathbb{E}_{\boldsymbol{a}_t \sim \pi} \sum_{t=1}^{T} \gamma^{t-1} r_t, \quad \forall \pi \in \Pi. \tag{A38}$$

Recall k is monotonically increasing, therefore, the order of the sequence of reward $[r_t] \mid_{t=1}^{T}$ is the same as $[k(r_t)] \mid_{t=1}^{T}$. That is, given any pairs of two reward sequences, $[r_t] \mid_{t=1}^{T}$ and $[r'_t] \mid_{t=1}^{T}$,

$$\sum_{t=1}^{T} \gamma^{t-1} r_t > \sum_{t=1}^{T} \gamma^{t-1} r'_t \rightleftharpoons \quad \sum_{t=1}^{T} \gamma^{t-1} k(r_t) > \sum_{t=1}^{T} \gamma^{t-1} k(r'_t). \tag{A39}$$

Observed above, we can deduce the following,

$$\arg\max_{\pi \in \Pi} \mathbb{E}_{\boldsymbol{a}_t \sim \pi} \sum_{t=1}^{T} \gamma^{t-1} r_t \rightleftharpoons \quad \arg\max_{\pi \in \Pi} \mathbb{E}_{\boldsymbol{a}_t \sim \pi} \sum_{t=1}^{T} \gamma^{t-1} k(r_t). \tag{A40}$$

Therefore, learning a policy that maximizes $V^\pi(s)$ calculated by r_t is equivalent to learning a policy that maximizes $V^\pi(s)$ calculated by $k(r_t)$.

G Experimental Details

(1) *Classic (Linear) MPE*. We evaluate our method in *Predator-Prey* scenarios, where the agent numbers vary in 3, 6 and 15. In the classic MPE, the team reward equals the sum of individual rewards. This characteristic provides benefits for those methods that assume a linear sum of individual rewards, like STAS [4]. What makes it challenging is the episodic modification: we mark the performance of the team as R_t for each timestep but only award they $\sum_{t=1}^{T} R_t$ when $t = T$, otherwise 0. (2) *Nonlinear MPE*. Classic MPE is easy to address for the previous work making a linear assumption of team reward, but not general. In order to create a more general evaluation platform, we modify the linear team reward setting in *classic MPE* into a nonlinear setting, posing obstacles in spatial credit assignment. We obtain the episodic rewards in the same way as that in the *classic MPE* as well. For more details, please refer to Appendix G. **Metrics.** We evaluate the effectiveness of credit assignment of all the methods by reporting the average accumulative reward across three random seeds. Intuitively, higher rewards indicate better performance of credit assignment algorithms.

G.1 Multi-agent Particle Environment (Predator-Prey)

We use Predator-Prey in our experiments, as illustrated in Figure A1. Good agents (green) are faster and are controlled by pre-trained model [21, 19]. Adversaries are slower and aim to hit good agents. Obstacles (large black circles) block the way. By default, there are N good agents, $3N$ adversaries, and several landmarks.

State Each agent observes an ego-centric state. For the agent n, the state consists of,

- agent.vel: agent n's x, y-axis velocity;
- agent.pos: agent n's x, y-axis location in the global world;
- landmark.relative_pos: the landmarks's relative x, y-axis position in the agent n's frame.
- good_agent.relative_pos: the other adversaries agent's relative x, y-axis position in the agent n's frame.
- adversary_agent.relative_pos: the good agent's relative x, y-axis position in the agent n's frame.
- other_agent.relative_vel: the good agent's velocity.

Reward Let there be $3N$ predators and N prey. We evaluate our method in the Predator-Prey scenarios of $3N \in [3, 6, 15]$. For each predator, the reward calculation is as follows:

1. Calculate the distance, $d_t^{i,j}$, between the predator i and the prey j
2. Denote the distance of predatory i to its nearest prey as $d_t^i = \min_j d_t^{i,j}$
3. The **individual rewards** for each predator, r^i, is given by $r^i = -0.1 \times d_t^i$.
4. If a collision occurs between the predator i and prey, the predator is award $+10$ *i.e.* $r^i = r^i + 10 \times \sum_j \mathbf{1}(d^{i,j}) < d_{\text{sh}})$, where d_{sh} is the threshold for collision and $\mathbf{1}$ is indicator function.
5. The **team reward** in the *classic MPE* is then given by $R_t = \sum_{i=1}^{3N} r_t^i$.
6. The **team reward** in the *Nonlinear MPE* is then given by $R_t = \sum_{i=1}^{3N} \tilde{r}_t^i$ where transformed individual reward $\tilde{r}^i$ is,

$$\tilde{r}^i = \begin{cases} \exp(\bar{r}_t) & \bar{r}_t < 0, \\ 5 \times [1 + \log(\bar{r}_t + 1)] & \bar{r}_t > 0, \end{cases} \tag{A41}$$

   and we provide python code for generating $\tilde{r}_t$ for different scenarios in Listing 1, Listing 2 and Listing 3.
7. When the episode ends, the **episodic reward** is given by $Q = \sum_{t=1}^{T} R_t$.

```python
    def nonlinear_mixture(self, rew):
        # maping from rew_i, rew_{i+1}:
        rew_i = copy.deepcopy(rew)
        rew_j = np.concatenate([rew_i[1:], rew_i[:1]])
        rew_jj = np.concatenate([rew_i[2:], rew_i[:2]])

        sum_rew_ij = 0.75 * rew_i + 0.5 * rew_j + 0.25 * rew_jj

        # convert into transformed rewards
        rew = np.piecewise(sum_rew_ij, [sum_rew_ij < 0, sum_rew_ij >= 0], [lambda x: np.exp(x), lambda x: 5 * (1 + np.log(x+1))])

        self.transformed_individual_rew = rew
        # return team reward
        return rew.sum()
```

Listing 1: Team reward in Nonlinear Predator Prey ($N = 3$)

```python
    def nonlinear_mixture(self, rew):
        # maping from rew_i, rew_{i+1}:
        rew_i = copy.deepcopy(rew)
        sum_rew_ij = 2 * rew_i
```

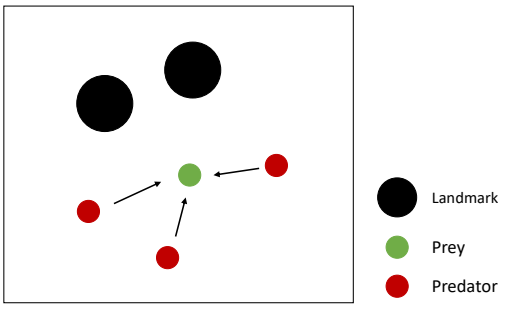

Figure A1: Predator-Prey scenario (3 agents) in Multi-agent Particle Environments.

```
        coef = 0.75
        for i in range(3):
            rew_j = np.concatenate([rew_i[i:], rew_i[:i]])
            sum_rew_ij += coef * rew_j
            coef *= 0.6

        # convert into transformed rewards
        rew = np.piecewise(sum_rew_ij, [sum_rew_ij < 0, sum_rew_ij >= 0], [lambda x: np.exp(x), lambda x: 5 * (1 + np.log(x+1))])
        self.transformed_individual_rew = rew
        return rew.sum()
```

Listing 2: Team reward in Nonlinear Predator Prey ($N = 6$)

```

    def nonlinear_mixture(self, rew):
        # maping from rew_i, rew_{i+1}:
        rew_i = copy.deepcopy(rew)

        sum_rew_ij = rew_i
        coef = 0.75
        for i in range(6):
            rew_j = np.concatenate([rew_i[i:], rew_i[:i]])
            sum_rew_ij += coef * rew_j
            coef *= 0.8

        # convert into transformed rewards
        rew = np.piecewise(sum_rew_ij, [sum_rew_ij < 0, sum_rew_ij >= 0], [lambda x: np.exp(x), lambda x: 5 * (1 + np.log(x+1))])
        self.transformed_individual_rew = rew
        return rew.sum()
```

Listing 3: Team reward in Nonlinear Predator Prey ($N = 15$)

It is important to note, that the above is the generative process of the individual rewards, transformed individual rewards, team rewards, and episodic rewards. We address the episodic reward setting, instead of observing the individual rewards and the team rewards, the agent can only observe the episodic rewards when the episode ends, otherwise, it gets zero.

G.2 Baselines

We compare our method with several baselines,

- QMIX [28] is a value-based Centralized Training with a Decentralized Execution (CTDE) approach that computes joint action values through a monotonic non-linear combination of individual agent values, which are based solely on local observations. This design enables efficient maximization of the joint action value in off-policy learning and ensures alignment between centralized and decentralized policies.

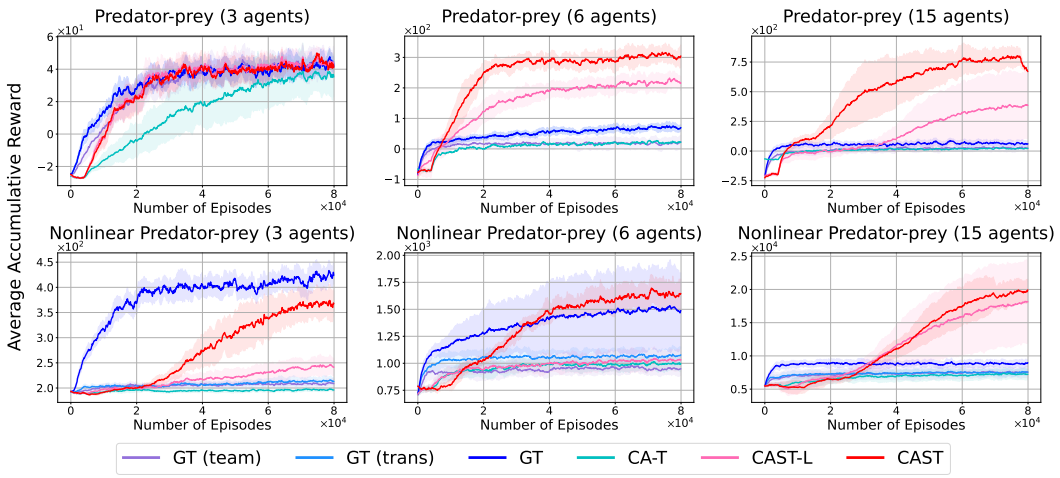

Figure A2: Ablation study on Multi-agent Particle Environment (MPE). The first row is the accumulative reward in classical MPE, while the second row is in the modified MPE.

- COMA [7], a Centralized Training with Decentralized Execution (CTDE) technique that effectively marginalizes an individual agent's action for evaluation: holding the actions of other agents fixed and comparing the estimated return for the joint action to the conterfactual baseline. Meanwhile, their critic architecture facilitates the swift computation of the counterfactual baseline within a single forward pass, enhancing efficiency.

- SQDDPG [41] addresses the inaccurate credit assignment and inefficient policy learning caused by the team reward. They extend the convex game (ECG) and design a local reward approach called Shapley Q-value to distribute the global reward, which reflects each agent's own contribution and serves as the critic for each agent.

- STAS [4] addresses the spatial-temporal credit assignment by learning the individual reward redistribution model for each agent. After the decomposition of the long-term return into each timestep, STAS uses Shapley values to redistribute the individual payoff of agents. However, STAS assumes that the long-term return is equal to the linear sum of individual rewards from all the agents and timesteps, which is relatively more strict than the assumption we made in this paper.

H Additional Experimental Results

Policy Learning with Ground Truth Rewards. As shown in Figure A2, we provide the results of policy learning with ground truth individual rewards (**GT**), transformed individual rewards (**GT (trans)**) and team rewards (**GT (team)**). We are surprised that the performance is improved using the estimated rewards. A possible reason is that such a reward can encourage exploration, thus leading to better policies [9].

Recovery Accuracy. We provide Spearman's rank correlation coefficient between recovered individual rewards and the ground truth individual rewards to demonstrate the accuracy of the recovery and the necessity of relaxing the linear assumption across the agents' individual contributions. According to Table A1, **CAST** achieves comparable with the **CAST-L** in the Classic MPE and achieves the best estimation of individual rewards in the Nonlinear MPE. Note that **CAST-L** in Classic MPE is the model with the prior knowledge of the linear team reward setting, which is expected to be the best one in the linear MPE. Therefore the comparable result demonstrates that our method can be regarded as a general method to address both linear and nonlinear team reward settings without the requirement of domain knowledge.

	Classic MPE			Nonlinear MPE		
	N=3	N=6	N=15	N=3	N=6	N =15
Ours	0.92	0.79	0.65	**0.84**	**0.74**	**0.73**
CAST-L	**0.94**	**0.84**	**0.74**	0.47	0.62	0.55
STAS	0.40	0.27	0.20	0.21	0.16	0.04

Table A1: Spearman's rank correlation coefficient between recovered individual rewards and the ground truth individual rewards. The best values are in **bold**.

I Other Implementation Details

I.1 Overall Pipeline

As shown in Algorithm 1, we train the generative model and the policy model alternately. In each epoch of generative model learning, we first train the transformed individual reward predictor Φ_{trans} and then train the iVAE Φ_{inv}.

Algorithm 1 Causally-inspired Spatial-Temporal Credit Assignment

Require: Environment $\mathcal{E}$, Generative Model $\phi_{\text{m}} := [\Phi_{\text{trans}}, \Phi_{\text{inv}}]$; Policy Model ϕ_π; Replay buffer $\mathcal{B}_{\text{m}}, \mathcal{B}_\pi \leftarrow \emptyset$; Frequency M.

for i in $n_{\text{training epoch}}$ **do**
  If using On-policy Algorithm: $\mathcal{B}_\pi \leftarrow \emptyset$
  for b in $n_{\text{batch size}}$ **do**
    Sample trajectory $\tau = [\boldsymbol{s}_t, [\boldsymbol{a}_t^n, \boldsymbol{o}_t^n]_{n=1}^N]_{t=1}^T \cup Q$ from $\mathcal{E}$, where Q is the long-term return
    store τ into the buffer $\mathcal{B}_{\text{m}}$, $\mathcal{B}_\pi$
  end for
  // Generative Model Learning
  if $i \mod M = 0$ **then**
    for ii in n_{steps} **do**
      Sample a batch of trajectories $\mathcal{D} \sim \mathcal{B}_{\text{m}}$
      Calculate L_{trans} through Eq. **??**
      Optimize: $\Phi_{\text{trans}} \leftarrow \Phi_{\text{trans}} - \alpha \nabla_{\Phi_{\text{trans}}}(L_{\text{trans}})$
    end for
    for ii in n_{steps} **do**
      Sample a batch of trajectories $\mathcal{D} \sim \mathcal{B}_{\text{m}}$
      Calculate L_{ELBO} through Eq. 5
      Optimize Φ_{inv}: $\Phi_{\text{inv}} \leftarrow \Phi_{\text{inv}} - \alpha \nabla_{\Phi_{\text{inv}}} L_{\text{ELBO}}$
    end for
  end if
  // Independent Policy Learning
  Sample data $\mathcal{D}$ from $\mathcal{B}_\pi$
  Calculate individual rewards r_t^n through Φ_{m}
  Calculate J_π using r_t^n
  Optimize the policy $\Phi_\pi \leftarrow \Phi_\pi - \alpha \nabla_{\Phi_\pi} J_\pi$
end for

I.2 Generative Model Learning

The generative model contains: Φ_{trans}, $[\Phi_{\text{inv}} := \phi_f, \phi_{\text{enc}}, \phi_g]$, whose network structure is listed in Table A2.

Below, we give an illustration of the $\phi_{\text{cau}} \in \mathbb{R}^{N \times |\boldsymbol{s}|}$. Recall the definition of $\boldsymbol{C}_i^n \in [0, 1]$: if $\boldsymbol{C}_i^n = 1$, then the corresponding causal edge exists, and the i-th dimension of joint state causes agent n's

Layer#	1	2	3	4
Φ_{trans}	FC256	FC128	FC1	-
ϕ_f	FC64	FC64	FC1	-
ϕ_{enc}	FC128	FC64	FC64	FC1
ϕ_g	FC64	FC64	FCN	-
ϕ_π	FC64	FC64	FC$\lvert\boldsymbol{a}\rvert$	-
ϕ_v	FC64	FC64	FC1	-

Table A2: The network structures used in CAST. CF256 denotes a fully-connected layer with an output size of 256. Each hidden layer is followed by an activation function, LReLU. $|\boldsymbol{a}|$ is the number of dimensions of the action in a specific task. N is the number of agents.

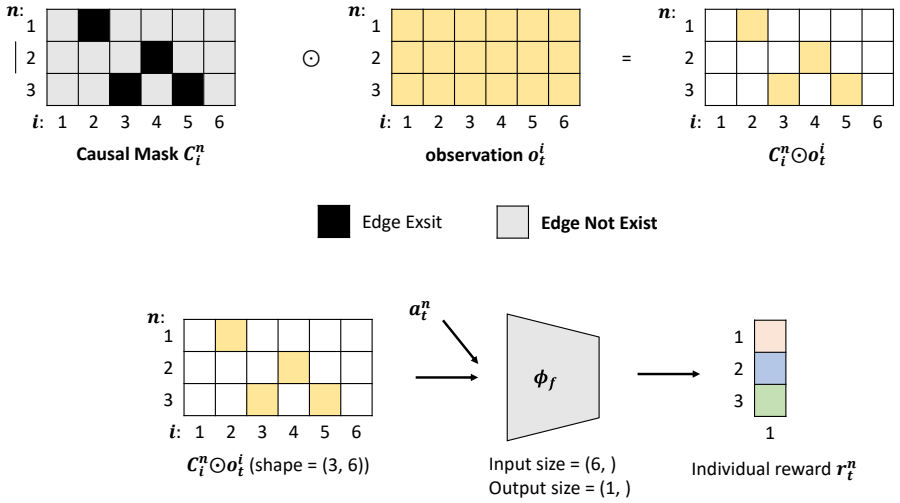

Figure A3: Illustration of using the causal mask to predict the reward by ϕ_f.

individual rewards. We define,

$$\boldsymbol{C}_i^n = \begin{cases} \text{Sigmoid}(\phi_{\text{cau}}^{n,i}) & \text{for training} \\ \text{Sigmoid}(\phi_{\text{cau}}^{n,i}) & \text{for inference, if Sigmoid}(\phi_{\text{cau}}^{n,i}) > 0.1 \\ 0 & \text{for inference, if Sigmoid}(\phi_{\text{cau}}^{n,i}) < 0.1 \end{cases} \tag{A42}$$

where $\phi_{\text{cau}}^{n,i}$ denotes the (n, i)-element in the free parameters ϕ_{cau}.

One example of using the estimated mask is given as Figure A3.

I.3 Policy Model Learning

Considering the specific requirements of the employed RL algorithm, Proximal Policy Optimization (PPO), our Policy Model Φ_π comprises two components, the actor ϕ_π and the critic ϕ_v. Detailed network structures for both components can be found in Table A2.

I.4 Hyper-parameters

The network is trained from scratch using the Adam optimizer, without any pre-training. The initial learning rate for Φ_{trans}, Φ_{inv} and Φ_π are set to 1×10^{-4}, 5×10^{-4} and 1×10^{-4}, separately. The hyperparameters for policy learning are shared across all tasks, with a discount factor of 1.00. To facilitate training, we utilize a replay buffer with a size of 2×10^6 time steps for policy learning and a size of 2×10^6 trajectories for generative model learning. The warmup size of the buffer for generative model learning is set to 1×10^3 timesteps and 4×10^3 timesteps. The model is trained for

Table A3: The hyper-parameters.

hyperparameters	value	hyperparameters	value
epochs	1000	optimizer	Adam
Φ_π learning rate	1×10^{-4}	Φ_{trans} learning rate	5×10^{-4}
Φ_{inv} learning rate	1×10^{-4}	Φ_π train_batches	100
Φ_{trans} train_batches	50	Φ_{inv} train_batches	60
Φ_π replay buffer size	2×10^6	Φ_{m} replay buffer size	2×10^6
Φ_{m} training frequency	3	evaluation episodes	30
γ	1.00	λ	1e-3

1000 epochs, with each epoch consisting of 100 policy learning cycles and 50 cycles for Φ_{trans} and 60 cycles for Φ_{inv}. During each iteration, we collect data from 2×10^3 time steps of interaction with the MPE simulation, which is then stored in the replay buffer. For training the Φ_{m}, we sample 512 trajectories, each is no more than 25 steps. As for policy learning and the optimization of ϕ_{dyn}, we use data from 2000 time steps. Validation is performed per training epoch, and the average metric is computed based on 30 test rollouts. The hyperparameters for learning the CAST model can be found in Table A3. All experiments were conducted on an HPC system equipped with 128 Intel Xeon processors operating at a clock speed of 2.2 GHz and 5 terabytes of memory. Runtime should be ranged between 5 hours to 22 hours for the different agent numbers in the evaluation environments. Our code is built based on Code.

NeurIPS Paper Checklist

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: we discuss the limitations in Section C

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide code, the hyperparameter setting as well as the used neural network structure.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We use the open-source Multi-agent Particle Environment and provide the code for the transformed reward function in Nonlinear Predator Prey in the Appendix. The training code is provided through Anonymous GitHub: Code.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the experimental details in Section 3 and Appendix I.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We plot the shaded region (standard deviation) in the Figure 3 and Figure A2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the information on the computer resources in the Appendix I.4

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    We provide the broader impacts of our work in Section D.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We use an open-source evaluation environment, and the code base is from Code.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original paper of Multi-agent Particle Environment in Section 3 and the GitHub link of the code that we built on in Appendix I.4.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    We do not involve crowdsourcing experiments and research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.