Individual Contributions as Intrinsic Exploration Scaffolds for Multi-agent Reinforcement Learning

Xinran Li [1] Zifan Liu [1] Shibo Chen [1] Jun Zhang [1]

Abstract

In multi-agent reinforcement learning (MARL), effective exploration is critical, especially in sparse reward environments. Although introducing global intrinsic rewards can foster exploration in such settings, it often complicates credit assignment among agents. To address this difficulty, we propose Individual Contributions as intrinsic Exploration Scaffolds (ICES), a novel approach to motivate exploration by assessing each agent's contribution from a global view. In particular, ICES constructs *exploration scaffolds* with Bayesian surprise, leveraging global transition information during centralized training. These scaffolds, used only in training, help to guide individual agents towards actions that significantly impact the global latent state transitions. Additionally, ICES separates exploration policies from exploitation policies, enabling the former to utilize privileged global information during training. Extensive experiments on cooperative benchmark tasks with sparse rewards, including Google Research Football (GRF) and StarCraft Multi-agent Challenge (SMAC), demonstrate that ICES exhibits superior exploration capabilities compared with baselines. The code is publicly available at `https://github.com/LXXXXR/ICES`.

1. Introduction

Multi-agent reinforcement learning (MARL) has recently gained significant interest in the research community, primarily due to its applicability across a diverse range of practical scenarios. Numerous real-world applications are multi-agent in nature, ranging from resource allocation (Ying & Dayong, 2005) and package logistics (Seuken & Zilberstein, 2007) to emergency response operations (Parker et al., 2016) and robotic control systems (Swamy et al., 2020). The multi-agent settings introduce unique challenges beyond the single agent reinforcement learning (RL), such as the need to address non-stationarity and partial observability (Yuan et al., 2023a), as well as the complexities involved in credit assignment (Foerster et al., 2018).

Despite the progress made by state-of-the-art algorithms like MADDPG (Lowe et al., 2017), QMIX (Rashid et al., 2020) and MAPPO (Yu et al., 2022), which leverage the centralized training decentralized execution (CTDE) paradigm, a significant limitation arises in environments with sparse rewards. Sparse rewards, common in real-world applications, present a substantial challenge for policy exploration as they provide limited guidance during training. Classical exploration methods, such as ϵ-greedy, struggle in these environments, primarily due to the exponentially growing state space and the necessity for coordinated exploration among agents (Liu et al., 2021). To address sparse rewards in MARL, recent approaches have focused on augmenting extrinsic rewards with global intrinsic rewards. These intrinsic rewards are typically designed to foster cooperation (Wang et al., 2019) or diversity (Li et al., 2021). While showing promise, these methods suffer from one obstacle: the non-stationary nature of intrinsic rewards during training (Burda et al., 2018) introduces additional complications in credit assignment. Furthermore, balancing intrinsic and extrinsic rewards often demands considerable tunning effort, particularly in the absence of prior knowledge about the extrinsic reward functions (Yuan et al., 2023b).

To address the aforementioned challenges and improve performance in MARL with sparse rewards, we propose a new exploration method, named Individual Contribution as Intrinsic Exploration Scaffolds (ICES). The key idea is to take advantage of the CTDE paradigm and utilize global information available during the training to construct intrinsic scaffolds that guide multi-agent exploration. These intrinsic scaffolds are specifically designed to encourage individual actions that have a significant influence on the underlying global latent state transitions, thus promoting cooperative exploration without the need to learn intrinsic

[1] Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China. Correspondence to: Shibo Chen <eeshibochen@ust.hk>.

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

credit assignment. Furthermore, these scaffolds, akin to physical scaffolds in construction, will be dismantled after training to prevent any effect on execution latency.

In particular, we make two key technical contributions. Firstly, we shift the focus from global intrinsic rewards to individual contributions as the primary motivation for agent exploration. This approach effectively circumvents the complexities involved in credit assignment for global intrinsic rewards. To encourage agents to perform cooperative exploration, we capitalize on the centralized training to estimate the Bayesian surprise related to agents' actions, quantifying their individual contributions. This is achieved by employing a conditional variational autoencoder (CVAE) with two encoders. Secondly, we optimize exploration and exploitation policies separately with distinct RL algorithms. In this way, the exploration policies can be granted access to privileged information, such as global observations, which helps to alleviate the non-stationarity challenge. Importantly, these exploration policies serve as temporary scaffolds, and do not intrude on the decentralized nature of the execution phase.

We evaluate the proposed ICES on two benchmark environments: Google Research Football (GRF) and StarCraft Multi-agent Challenge (SMAC), under sparse reward settings. The empirical results and comprehensive ablation studies demonstrate ICES's superior exploration capabilities, notably in convergence speed and final win rates, when compared with existing baselines.

2. Background

In this section, we briefly introduce the fully cooperative multi-agent task considered in this work and provide essential background on CVAEs, which will be utilized for constructing meaningful individual contribution assessment. Then, we provide an overview of related works on multi-agent exploration.

2.1. Problem Setting

Decentralized Partially Observable Markov Decision Process (Dec-POMDP): We consider a fully cooperative partially observable multi-agent task modeled as a decentralized partially observable Markov decision process (Dec-POMDP) (Oliehoek & Amato, 2016). The Dec-POMDP is defined by a tuple $\mathcal{M} = \langle \mathcal{S}, U, P, R, \Omega, O, n, \gamma \rangle$, where n denotes the number of agents and $\gamma \in (0, 1]$ is the discount factor that balances the trade-off between immediate and long-term rewards.

At timestep t, with the global observation $s \in \mathcal{S}$, agent i receives a local observation $o_i \in \Omega$ drawn from the observation function $O(s, i)$. Subsequently, the agent selects an action $u_i \in U$ based on its local policy π_i. These individual actions collectively form a joint action $\boldsymbol{u} \in U^n$, leading to a transition to the next global observation $s' \sim P(s'|s, \boldsymbol{u})$ and yielding a global reward $r = R(s, \boldsymbol{u})$. For clarity, we refer to this global reward as the extrinsic reward r_{ext}, distinguishing it from the agents' intrinsic motivations. Each agent keeps a local action-observation history denoted as $h_i \in (\Omega \times U)$. The team objective is to learn the policies that maximize the expected discounted accumulated reward $G_t = \sum_t \gamma^t r^t$.

2.2. Conditional Variational Autoencoders (CVAEs)

CVAEs (Sohn et al., 2015) extend variational autoencoders (VAEs) to model conditional distributions, adept at handling scenarios where the mapping from input to output is not one-to-one, but rather one-to-many (Sohn et al., 2015). The generation process in a CVAE is as follows: given an observation $\mathbf{x}$, a latent variable $\mathbf{z}$ is sampled from the prior distribution $p_\theta(\mathbf{z}|\mathbf{x})$, and the output $\mathbf{y}$ is generated from the conditional distribution $p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})$. The objective is to maximize the conditional log-likelihood, which is intractable in practice. Therefore, the variational lower bound is maximized instead, which is expressed as:

$$\begin{aligned}\mathcal{L}_{\text{CVAE}}(\mathbf{x}, \mathbf{y}; \theta, \phi) = D_{\text{KL}}\left[q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) \parallel p_\theta(\mathbf{z}|\mathbf{x})\right] \\ + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p(\mathbf{y}|\mathbf{x}, \mathbf{z})\right],\end{aligned} \tag{1}$$

where D_{KL} denotes the Kullback–Leibler (KL) divergence and $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$ is an approximation of the true posterior.

2.3. Related Works

Besides adapting exploration techniques from single-agent RL, considerable research has focused on developing exploration methods tailored to multi-agent settings. We categorize these efforts into two broad types: global-level and agent-level exploration.

Global-level Exploration: Research in this domain aims to encourage exploration in the global space. For instance, MAVEN (Mahajan et al., 2019) integrates hierarchical control by introducing a latent space to guide exploration. Studies by Liu et al. (2021), Xu et al. (2023b) and Jo et al. (2024) focus on reducing the exploration space by identifying key subspaces. MAGIC (Chen et al., 2022) adopts goal-oriented exploration for multi-stage tasks. Other approaches emphasize encouraging desirable collective behaviors, such as the work by Chitnis et al. (2019) that emphasizes fostering synergistic behaviors among agents, and LAIES (Liu et al., 2023a), which avoids lazy agents by encouraging diligence. Additionally, methods like EMC (Zheng et al., 2021) and MASER (Jeon et al., 2022) seek to enhance sample efficiency by effectively utilizing existing experiences in replay buffers, either by replaying high-reward sequences or creating subgoals for cooperative exploration.

Agent-level Exploration: This category of research incorporates specific objectives at the individual agent level. EITI and EDTI (Wang et al., 2019) focus on maximizing the mutual influence among agents' state transitions and values. Li et al. (2021) promotes diverse behaviors among agents and Xu et al. (2023a) proposes to encourage diverse joint policy compared to historical ones. SMMAE (Zhang et al., 2023) fosters individual curiosity, while ADER (Kim & Sung, 2023) introduces an adaptive entropy-regularization scheme to allow varied levels of exploration across agents.

While global-level exploration aids in fostering cooperative behaviors, the integration of global extrinsic and intrinsic rewards often complicates credit assignment, potentially hindering algorithm performance. Some methods (Chen et al., 2022; Liu et al., 2023a) also rely on parsing the global observation, and thus require specific domain knowledge, which thereby limits their applicability. In contrast, agent-level exploration offers a more straightforward approach but may result in less coordinated actions among agents. Our method seeks to combine the advantages of both approaches, assigning specific motivations to individual agents while leveraging global information to shape these motivations.

Beyond the literature on exploration, we discuss two other research lines that share similar techniques to those used in this work:

Intrinsic Rewards for More than Exploration: In addition to leveraging intrinsic rewards for better exploration, the concept of intrinsic motivation has been applied to other aspects of MARL. For example, LIIR (Du et al., 2019) proposes utilizing intrinsic rewards to explicitly assign credits to different agents, resulting in an algorithm of enhanced performance. Other works design intrinsic rewards to incorporate preferences such as social influences (Jaques et al., 2019), social diversity (McKee et al., 2020), and alignment(Ma et al., 2022; Zhang et al., 2024) into the learned policies.

Credit Assignment: Credit assignment is a key challenge in MARL, referring to how to allocate global rewards to provide accurate feedback for individual agents (Yuan et al., 2023a). Several value decomposition methods, such as VDN (Sunehag et al., 2018), QMIX (Rashid et al., 2020), and QTRAN (Son et al., 2019), have been proposed to implicitly assign credits among agents for discrete actions, and a later work LICA (Zhou et al., 2020) tackles the same issue in the continuous action domain. COMA (Foerster et al., 2018) takes a different approach and uses the counterfactual baseline to explicitly measure each agent's contribution. More recently, NA2Q (Liu et al., 2023b) proposes an interpretable credit assignment framework by exploiting generalized additive models. Unlike these works that aim to solve the credit assignment challenge, our work focuses on avoiding extra complexity brought by non-stationary intrinsic rewards to the original credit assignment problem.

3. Individual Contributions as Intrinsic Exploration Scaffolds

In this work, we propose a novel approach of leveraging individual contributions as intrinsic scaffolds to enhance exploration in MARL. It aims to fully utilize privileged global information during centralized training while ensuring decentralized execution remains unaffected.

The following subsections will address three key questions: 1) **Why use individual contributions as intrinsic scaffolds?** Section 3.1 examines the advantages of focusing on the contributions of individual agents over collective team efforts in enhancing exploration strategies within MARL. 2) **How to assess individual contributions?** Section 3.2 describes how our methods quantify each agent's impact on global latent state transitions using Bayesian surprise. 3) **How are these scaffolds utilized effectively?** Section 3.3 elaborates on how exploration and exploitation policies are optimized with distinct objectives and strategies to utilize these scaffolds effectively, thereby enhancing exploration without compromising the original training objectives or the decentralized execution strategy.

3.1. From Global Intrinsic Rewards to Individual Contributions

Previous methods largely rely on formulating a global intrinsic reward, which is then added to extrinsic rewards to incentivize agents to explore. This strategy presents two notable drawbacks: Firstly, adding intrinsic rewards to the existing extrinsic rewards alters the original training objective. Intrinsic rewards, often learned and thus non-stationary throughout the training phase (Burda et al., 2018), will introduce additional non-stationarity into the training objective.

Secondly, like global extrinsic rewards, global intrinsic rewards require credit assignment among agents, a task that becomes more challenging with the non-stationary nature of intrinsic rewards. These complications can be effectively bypassed by directly providing agents with individual intrinsic motivations. In our method, this is achieved by utilizing privileged global information available only during the centralized training phase, thus addressing the issues of non-stationarity and complex credit assignment.

This is further verified by empirical ablation studies in Section 4.3.

3.2. Assessing Individual Contributions to Construct Intrinsic Scaffolds

Bayesian Surprise to Characterize Individual Contributions: In this subsection, we assess the individual contribution, denoted as $r^i_{t,\text{int}}$, of a specific action u^i_t executed by agent i. The objective is to evaluate the impact of action u^i_t on the global latent state transitions.

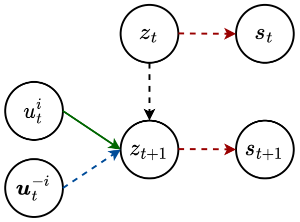


Figure 1: Dynamics model. The variable z denotes the latent state. The solid line indicates the individual contributions of agent i's actions, highlighted by the green arrow, signifying the primary focus of our measurement. Dashed lines represent other influences on state uncertainties, including the actions of other agents (blue arrow), which are excluded from agent i's contribution assessment, and the environment's inherent stochasticity (red arrows), known as the noisy TV problem, which we aim to mitigate.

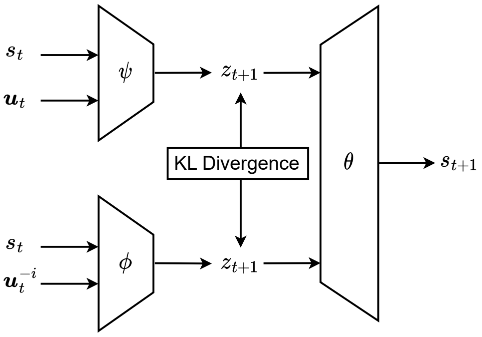


Figure 2: Modules for Bayesian surprise estimation. The structure resembles the CVAE structure with separate encoders and a shared decoder. The KL divergence between estimated priors is used as intrinsic contribution measurements.

To achieve this, we first demonstrate the environment dynamic model in Figure 1, illustrating how state transition uncertainties are influenced by several factors. These include the impact of agent i's action, represented by a mutual information term $r^i_{t,\text{int}} = I(z_{t+1}; u^i_t | s_t, \boldsymbol{u}^{-i}_t)$; the influence of other agents' actions, denoted by $I(z_{t+1}; \boldsymbol{u}^{-i}_t | s_t, u^i_t)$; and the inherent environmental uncertainties, expressed as the entropy $H(s_t|z_t)$, known as the noisy TV problem (Schmidhuber, 2010). Our focus is primarily on the individual contribution $r^i_{t,\text{int}}$, which necessitates a specific measurement method to effectively distinguish the contribution of agent i's action u^i_t and mitigate potential misattributions among agents and the effects of noisy TV. Consequently, we employ the Bayesian surprise as the measurement method. Following previous works (Itti & Baldi, 2005; Mazzaglia et al., 2022), we express the contribution $r^i_{t,\text{int}}$ as the mutual information between the latent variable z_{t+1} and the action u^i_t, which is given as

$$\begin{aligned} r^i_{t,\text{int}} &= I(z_{t+1}; u^i_t | s_t, \boldsymbol{u}^{-i}_t) \\ &= D_{\text{KL}}\left[p(z_{t+1}|s_t, \boldsymbol{u}_t) \parallel p(z_{t+1}|s_t, \boldsymbol{u}^{-i}_t)\right]. \end{aligned} \tag{2}$$

This term captures the discrepancy between the actual and counterfactual latent state distributions from the perspective of an individual agent i. In later sections, we omit the subscript t where contextually clear, referring to $r^i_{t,\text{int}}$ simply as r^i_{int}.

CVAE to Estimate the Bayesian Surprise: For robust estimation of individual contributions, it is essential to identify a latent space for z_t that is both compact and informative, capable of reconstructing the original state space. To achieve this, we resort to CVAE owing to its ability to induce explicit prior distributions and perform probabilistic inference, a necessity in environments with inherent stochasticity, such as GRF (Kurach et al., 2020).

We aim to estimate two specific priors: $p_\psi(z_{t+1}|s_t, \boldsymbol{u}_t)$ and $p_\phi(z_{t+1}|s_t, \boldsymbol{u}^{-i}_t)$. However, learning these two priors independently will not yield satisfactory results, as shown later in our ablation studies (Figure 7). This is due to the potential misalignment of the latent spaces created by each independently trained priors, rendering the KL-divergence measure less effective. Thus, to align the latent spaces, we use two separate encoders for these estimations while utilizing a shared decoder for reconstruction.

The CVAE's architecture, depicted in Figure 2, includes the following components:

Prior Encoders:	$p_\psi(z_{t+1}\|s_t, \boldsymbol{u}_t)$,
	$p_\phi(z_{t+1}\|s_t, \boldsymbol{u}^{-i}_t)$,
Reconstruction Decoder:	$p_\theta(s_{t+1}\|z_{t+1})$,
Latent Posteriors:	$q_\psi(z_{t+1}\|s_t, \boldsymbol{u}_t, s_{t+1})$,
	$q_\phi(z_{t+1}\|s_t, \boldsymbol{u}^{-i}_t, s_{t+1})$.

Training Objective for Scaffolds: The training objective of the above modules is to maximize the variational lower bound of the conditional log-likelihood (Sohn et al., 2015), formalized as:

$$\begin{aligned} \mathcal{J}(\psi, \phi, \theta) = &-D_{\text{KL}}\left[q_\psi(z_{t+1}|s_t, \boldsymbol{u}_t, s_{t+1}) \| p_\psi(z_{t+1}|s_t, \boldsymbol{u}_t)\right] \\ &- D_{\text{KL}}\left[q_\phi(z_{t+1}|s_t, \boldsymbol{u}^{-i}_t, s_{t+1}) \parallel p_\phi(z_{t+1}|s_t, \boldsymbol{u}^{-i}_t)\right] \\ &+ \mathbb{E}_{z\sim q_\psi}\left[\log p_\theta(s_{t+1}|z)\right] + \mathbb{E}_{z\sim q_\phi}\left[\log p_\theta(s_{t+1}|z)\right]. \end{aligned} \tag{3}$$

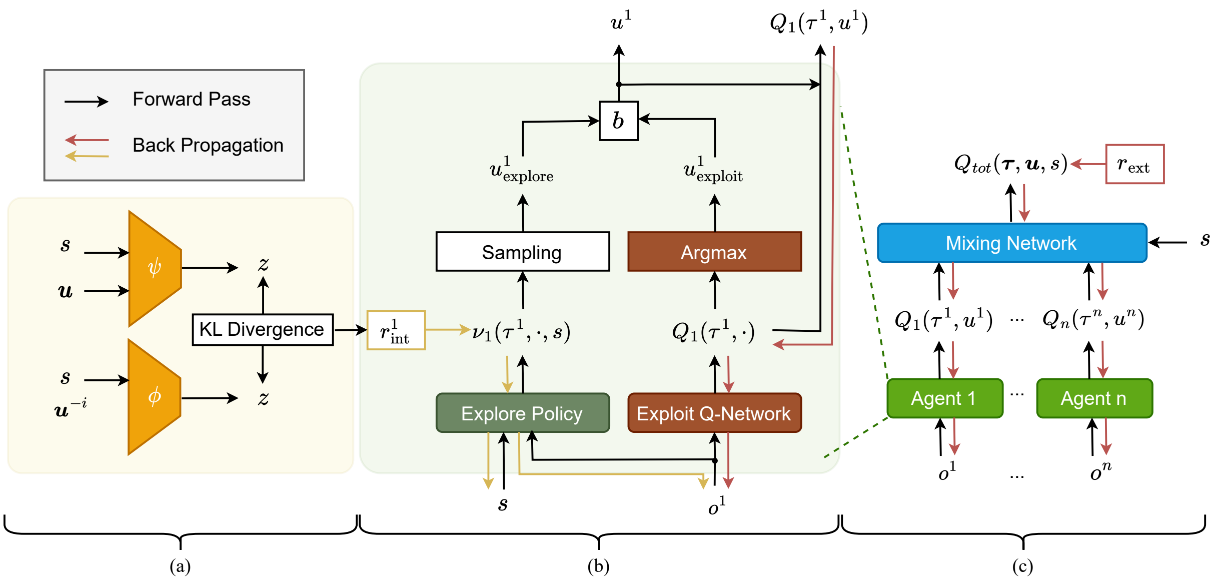


Figure 3: Network architecture of ICES. (a) Intrinsic exploration scaffolds. (b) Agent architecture. (c) Overall architecture. The red arrows denote the gradient flows guided by the global extrinsic reward r_{ext}, and the yellow arrows denote the gradient flows guided by the individual scaffolds r^i_{int}. The training objectives for the exploration policy and exploitation policy are decoupled while both policies are combined for action selection during the training phase.

3.3. Decoupling Exploration and Exploitation Policies to Utilize Intrinsic Scaffolds

In the ICES framework, we retain the training objective of learning the target (exploitation) policy $\boldsymbol{\pi}$, aiming at maximizing the cumulative reward $G_t = \sum_t \gamma^t r^t$. Concurrently, we adjust the behavior policy $\boldsymbol{b} = \{b_i\}_{i=1}^n$ to enhance exploration. Unlike the classical ϵ-greedy method adopted by most of the off-policy works, where actions are uniformly sampled if not following the target policy, ICES agents prioritize actions that significantly contribute to state transitions, as identified by r^i_{int}. The overall architecture is depicted in Figure 3, with different gradient flows denoted by red and yellow arrows for global extrinsic rewards and individual scaffolds, respectively.

We denote the target policy as $\boldsymbol{\pi} = \{\pi_i\}_{i=1}^n$, the exploration policy as $\{\nu_i\}_{i=1}^n$ and the behavior policy derived from the above two policies as $\boldsymbol{b} = \{b_i\}_{i=1}^n$.

Combining the Exploration and Exploitation Policies for Behavior Policies: As shown in part (b) of Figure 3, action selection involves both the exploration policy ν_i and the exploitation policy π_i. The behavior policy b_i is determined as follows:

$$u^i \sim b_i\left(u^i_{\text{explore}}, u^i_{\text{exploit}}\right) = \begin{cases} u^i_{\text{explore}} & \text{with probability } \alpha \\ u^i_{\text{exploit}} & \text{with probability } 1-\alpha \end{cases}, \tag{4}$$

where α is a hyperparameter balancing exploration and exploitation during training. The exploration and exploitation actions are given as:

$$u^i_{\text{explore}} \sim \nu_i(\tau^i, u, s), \tag{5}$$

$$u^i_{\text{exploit}} = \arg\max_u Q_i(\tau^i, u), \tag{6}$$

with $Q_i(\tau^i, \cdot)$ representing the local Q-value function for exploitation.

Optimization Objectives: The optimization objective for the exploitation policy, parameterized by ζ, is to minimize the TD-error loss:

$$\mathcal{L}(\zeta) = \mathbb{E}_{(\boldsymbol{\tau}_t, \boldsymbol{u}_t, s_t, r_{\text{ext}}, s_{t+1}) \sim \mathcal{D}} \left[\left(y^{tot} - Q_{tot}(\boldsymbol{\tau}_t, \boldsymbol{u}_t, s_t; \zeta) \right)^2 \right], \tag{7}$$

where $y^{tot} = r_{\text{ext}} + \gamma \max_{\boldsymbol{u}} Q_{tot}(\boldsymbol{\tau}_{t+1}, \boldsymbol{u}, s_{t+1}; \zeta^-)$ and ζ^- are the parameters of a target network as in DQN.

The optimization objective for the exploration policy, parameterized by ξ, is to maximize the average individual intrinsic scaffolds (not the episodic return objective) and exploration policy entropy:

$$\mathcal{J}_i(\xi) = \mathbb{E}_{\nu_i}[r^i_{\text{int}}] + \beta \mathcal{H}(\cdot|\tau^i, s), \tag{8}$$

where β is a hyperparameter to control the regularization weight for entropy maximization and $\mathcal{H}(\cdot|\tau^i, s) = -\mathbb{E}_{\nu_i(\xi)} \ln \nu_i(\cdot|\tau^i, s; \xi)$ is the entropy of policy ν_i at local-global observation pair (τ^i, s).

This approach ensures that while the training of the exploitation network remains centralized and retained, the exploration network benefits from decentralized training guided by individual intrinsic scaffolds. This strategy circumvents the challenges in intrinsic credit assignment. Moreover, since exploration policies are employed only during training, they can utilize privileged information, such as the global observation s, for more informed decision-making.

REINFORCE with Baseline for Exploration Policy Training: By decoupling the exploration and exploitation, we can employ distinct RL algorithms to update each policy, leveraging their respective strengths. For the exploitation policy update (denoted by the red arrows in Figure 3), we follow the previous work and use the DQN update with value decomposition methods like QMIX (Rashid et al., 2020) or QPLEX (Wang et al., 2020).

For the exploration policy, whose gradient is denoted by the yellow arrows in Figure 3, we prefer stochastic policies over deterministic ones for more diverse behaviors. Thus, we adopt a policy-based reinforcement learning algorithm with entropy regularization with the objective function given in Equation (8). To stabilize training, we introduce a value function $V(\tau^i, s; \eta)$ as a baseline. With the policy gradient theorem (details elaborated in Appendix A.1), we arrive at

$$\nabla_\xi \mathcal{J}_i(\xi) = \mathbb{E}_{\nu_i(\xi),(\tau_i,s)\sim\mathcal{D}}\left[A \cdot \nabla_\xi \ln \nu(\cdot|\tau^i, s; \xi)\right], \quad (9)$$

where $A = r^i_{\text{int}} - V(\tau^i, s; \eta) - \beta$ is the advantage function and $V_\eta(\tau^i, s)$ is updated by minimizing

$$\mathcal{L}(\eta) = \mathbb{E}_{\nu_i(\xi),(\tau_i,s)\sim\mathcal{D}}\left[\left(r^i_{\text{int}} - V(\tau^i, s; \eta)\right)^2\right]. \quad (10)$$

3.4. Overall ICES Training Algorithm

We summarize the overall training procedure for ICES in Algorithm 1. In particular, we train a scaffolds network (updated by Algorithm 3 in Appendix A.2) with parameters ψ, ϕ, θ and two policy networks (updated by Algorithm 2 in Appendix A.2), including an exploration network parametrized by ξ, η and an exploitation network parameterized by ζ. We utilize the scaffolds network to provide guidance for exploration network updates, and we utilize the exploration network to influence the action selection processes, consequently influencing the learning process of exploitation networks. Among the above networks, only the exploitation network will be used for execution.

Algorithm 1 Training Procedure of ICES

```
Init: Scaffolds parameters $\psi, \phi, \theta$
Init: Exploration networks parameters $\xi, \eta$
Init: Exploitation networks parameters $\zeta$
Init: $\mathcal{D} = \emptyset$, step $= 0$, $\theta^- = \theta$
while step $<$ step$_{\text{max}}$ do
  $t = 0$. Reset the environment.
  for $t = 1, 2, ...,$ episode_limit do
    for $i = 1, 2, ..., n$ do
      Select actions $u^i_t \sim b_i$            {▷ Equation (4)}
    end for
    $(s_{t+1}, \boldsymbol{o}_{t+1}, r_{t,\text{ext}}) =$ env.step$(\boldsymbol{u}_t)$
    $\mathcal{D} = \mathcal{D} \cup (s_t, \boldsymbol{o}_t, \boldsymbol{u}_t, r_{t,\text{ext}}, s_{t+1}, \boldsymbol{o}_{t+1})$
  end for
  if step $\mod$ train_interval $== 0$ then
    $\xi, \eta, \zeta \leftarrow$ TrainPolicies$(\psi, \phi, \xi, \eta, \zeta, \mathcal{D})$
                                        {▷ Algorithm 2 }
    $\psi, \phi, \theta \leftarrow$ TrainScaffolds$(\psi, \phi, \theta, \mathcal{D})$
                                        {▷ Algorithm 3 }
  end if
  if step $\mod$ target_update_interval $== 0$ then
    $\theta^- = \theta$
  end if
end while
Output: Exploitation networks parameters $\zeta$
```

4. Experiments

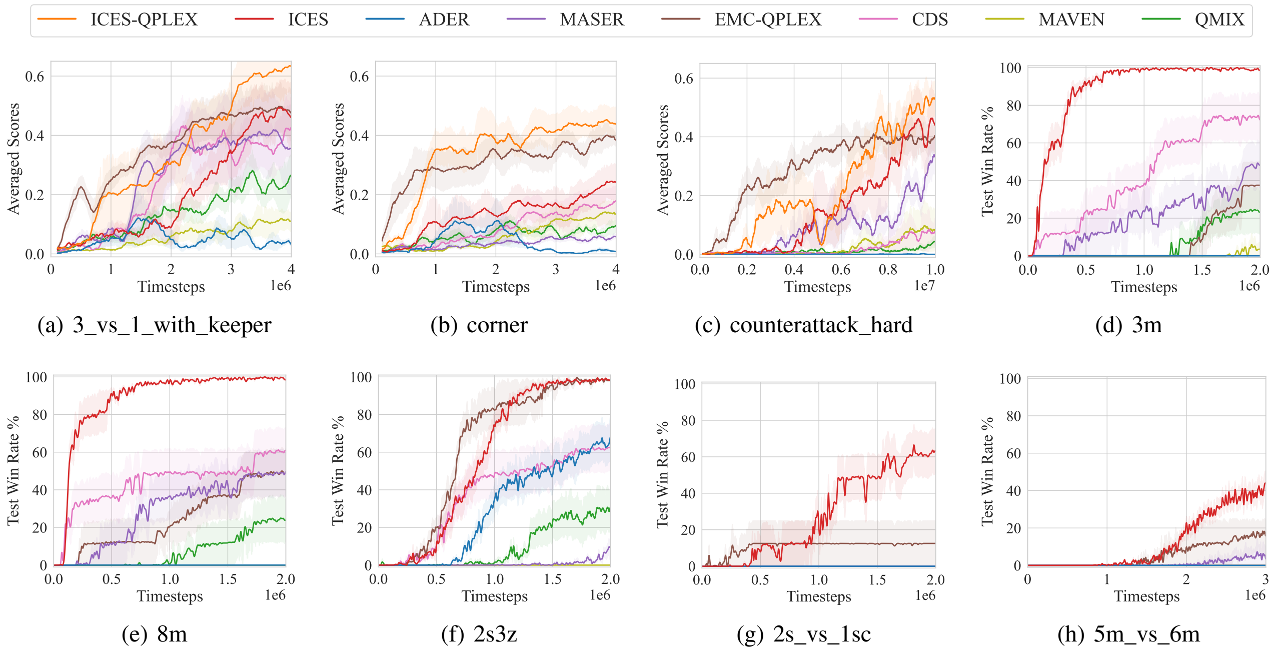
In this section, we evaluate ICES on two multi-agent benchmark tasks: GRF and SMAC. Experiments in the GRF domain are averaged over eight random seeds and experiments in the SMAC domain are averaged over five random seeds. The shaded areas represent 50% confidence intervals. Detailed descriptions of network architectures and training hyperparameters are available in Appendix B. Further experimental results are provided in Appendix C.

4.1. Settings

Benchmarks: In this work, we test ICES and baselines on widely used benchmarks of GRF and SMAC [1] in sparse reward settings, with details elaborated in Appendix B.2.

Baselines: We implement our proposed ICES on top of QMIX (Rashid et al., 2020). For the GRF environment, we add a curve combining ICES with QPLEX (Wang et al., 2020) and denote the results as ICES-QPLEX. We compare ICES with six state-of-arts baselines: ADER (Kim & Sung, 2023), MASER (Jeon et al., 2022), EMC (Zheng et al., 2021) (built upon QPLEX, denoted as EMC-QPLEX), CDS (Li et al., 2021), MAVEN (Mahajan et al., 2019) and QMIX (Rashid et al., 2020). Wherever possible, we utilize the official implementations of these baselines from their respective papers; in cases where the implementation is not available, we closely follow the descriptions provided in the papers and implement them on top of QMIX.

[1] We use SC2.4.10 with difficulty of 7. Note that performance is not always comparable across versions.



(a) 3_vs_1_with_keeper (b) corner (c) counterattack_hard (d) 3m

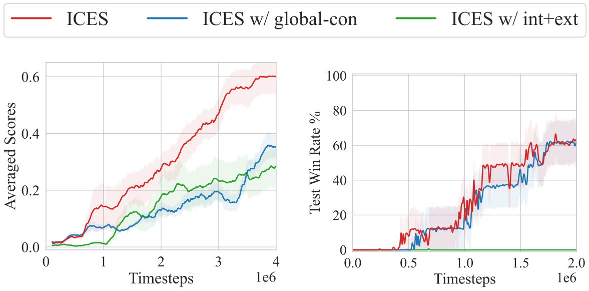
(e) 8m (f) 2s3z (g) 2s_vs_1sc (h) 5m_vs_6m

Figure 4: Performance comparison with baselines on GRF and SMAC benchmarks in sparse reward settings.

4.2. Benchmark results on GRF and SMAC

We present the comparative performance of ICES and various baselines in Figure 4. Overall, ICES demonstrates superior performance over baselines constructed on QMIX. When integrated with QPLEX (ICES-QPLEX), it surpasses baselines built on QPLEX. This showcases that ICES is able to foster effective exploration in MARL training, without tampering with the original training objective (discussed in Section 3.3), thus leading to a fast convergence and enhanced final performance. Challenges in GRF, including environmental stochasticity and the need for agent collaboration, are adeptly addressed by ICES. In particular, ICES filters out environmental noise using Bayesian surprise and promotes team coordination by constructing scaffolds based on global state transitions. (as discussed in Section 3.2.) It is also worth mentioning that, among the baselines, EMC-QPLEX also shows notable exploration capabilities, particularly in the early stages of training. This suggests that utilizing episodic memory, as EMC-QPLEX does, could be a beneficial approach to improve sample efficiency, albeit different from our focus on generating more informative exploration experiences.

For SMAC tasks, ICES also demonstrates strong performance compared with baselines, where most of the baselines require more training budget to find the winning strategy. For example, in scenario 2s_vs_1sc, ICES starts to observe winning episodes while baselines have not after 2M timesteps. In SMAC, the exploration challenges mainly arise from the large state space, which ICES addresses by computing intrinsic scaffolds within a compact latent space, rather than the extensive original state space.



(a) 3_vs_1_with_keeper (b) 2s_vs_1sc

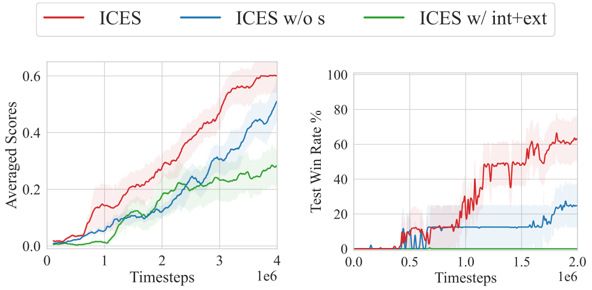
Figure 5: Ablations on individual contributions.

4.3. Ablation Studies

We conduct three sets of ablation studies regarding different aspects of our proposed ICES with one representative task from each benchmark task.

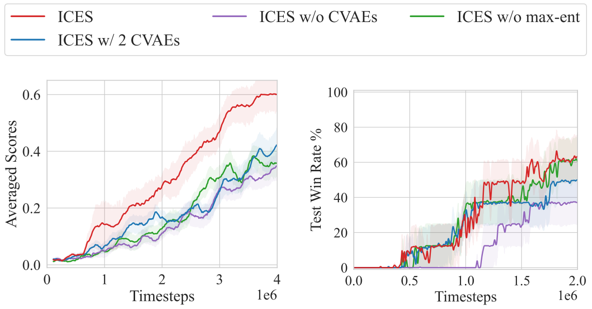
The first set of ablation studies focuses on individual contributions. We explore two distinct modifications to our original approach with the results presented in Figure 5:

- **ICES w/ global-con:** Instead of using the individual contribution as scaffolds for the corresponding agent following Equation (2), we use the collective contribution of all agents as a global scaffold. Here, the contribution of all agents is



(a) 3_vs_1_with_keeper (b) 2s_vs_1sc

Figure 6: Ablations on decoupling exploration and exploitation policies.



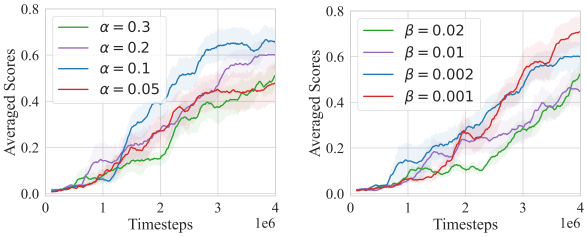
(a) 3_vs_1_with_keeper (b) 2s_vs_1sc

Figure 7: Ablations on other design choices.

estimated by $r_{t,\text{int}} = I(z_{t+1}; \boldsymbol{u}_t | s_t)$.
- **ICES w/ int+ext:** In this variant, individual scaffolds are directly summed up and used as global intrinsic rewards, similar to previous methods (Li et al., 2021).

Results in Figure 5 indicate that replacing individual contributions with global contributions hinders effective exploration. Notably, in the `3_vs_1_with_keeper` task, ICES achieves the final performance of over 60% winning rate while ICES w/ global-con only archives 40%. This highlights the misallocation of exploration incentives when contributions are not individually attributed, particularly in scenarios where specific agents play pivotal roles (e.g. the player with the ball in GRF). Moreover, further integrating the scaffolds into a global intrinsic reward exacerbates performance degradation. This could be attributed to the added complexity of needing to assign credit among agents of this new, non-stationary intrinsic reward, complicating the training process. Thus, this set of ablation studies underscores the effectiveness of directly assigning individual scaffolds to agents.

The second set of ablation studies investigates the effect of decoupling exploration and exploitation policies. We conducted experiments with two variants with the results shown in Figure 6:



(a) Analysis on α (b) Analysis on β

Figure 8: Hyperparameter analysis on the 3_vs_1_with_keeper scenario.

- **ICES w/o s:** This variant, diverging from the approach specified in Equation (5), excludes the global observation s from the exploration policy's inputs.
- **ICES w/ int+ext:** In this variant, individual scaffolds are directly summed up and used as global intrinsic rewards, similar to previous methods (Li et al., 2021).

Figure 6 shows the detrimental impact on exploration effectiveness when excluding global observation information from the exploration policies (while maintaining separate networks for exploration and exploitation). This highlights the significance of utilizing privileged information in exploration policies. Particularly in scenarios with pronounced partial observability, such as those encountered in the SMAC tasks, the lack of global information heavily deteriorates the exploration, with ICES achieving a final winning rate of 60% while ICES w/o s only achieving 20%. Furthermore, the performance further degrades when exploration and exploitation policies are merged into a single network. This set of ablation studies emphasizes the critical role of decoupling exploration and exploitation policies.

The last set of ablations are for other design choices in ICES and the results are given in Figure 7:

- **ICES w/o max-ent:** This variation eliminates the entropy regularization by by setting $\beta = 0$ in Equation (8).
- **ICES w/o CVAEs:** Contrary to leveraging KL-divergence in the latent space as stated in Equation (2), this variant directly calculates the intrinsic contribution as the Euclidean distance in the original state space.
- **ICES w/ 2 CVAEs:** Instead of employing two encoders and a shared decoder, this variant trains two independent CVAEs for Bayesian surprise estimation.

Figure 7 indicates that each of these modifications leads to a decline in performance in terms of final performance or convergence speed.

4.4. Hyperparameter Analysis

We further investigate the effect of different hyperparameters on the performance of ICES, as shown in Figure 8. The hyperparameter α controls the tradeoff between the

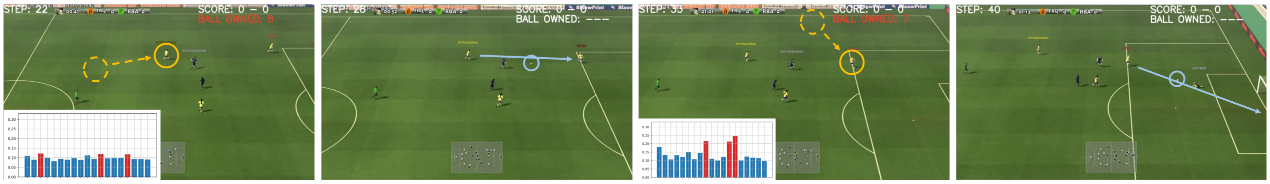


Figure 9: Visualization in the counterattack_hard scenario. We visualize some key frames of trained policies, with our team in yellow. The yellow dashed arrows denote the player movement while the blue arrows denote the ball movement. On the left bottom corners, we visualize the intrinsic scaffolds of the agent holding the ball, and red bars denote actions encouraged by intrinsic scaffolds.

exploration policy and exploitation policy, while β determines the balance between random exploration and directed exploration. Overall, within a reasonable range, ICES performs competitively across different hyperparameter settings, showcasing its robustness. However, achieving optimal performance requires proper tuning based on the specific task at hand.

4.5. Visualization

We visualize the final trained policies alongside some intrinsic scaffolds of the `counterattack_hard` scenario in Figure 9. We observe that on the first few timesteps, player 8, who possesses the ball, moves towards the right, aiming to approach its teammate palyer 7. At the same time, one of the highest rewarded actions identified by the intrinsic scaffolds is *short_pass*, which is beneficial because player 7 is closer to the goal. Consequently, guided by this intrinsic scaffold, player 8 executes a pass to player 7. Subsequently, player 7 receives the pass and makes a shot, resulting in a goal. Notably, right before the goal, player 7 is encouraged to *shoot* or *sprint*, both of which are good action candidates. This visualization result showcases how intrinsic scaffolds serve as a guiding mechanism, directing agents towards actions that are both exploratory and strategically sound.

5. Conclusions

In this work, we investigate MARL with sparse rewards. To facilitate cooperative exploration among agents without tampering the training objective, we propose ICES. Its key idea is to use estimations of individual contributions to encourage agents to choose actions that have more significant impact on the latent state transition during training time. ICES offers two main benefits: Firstly, with the individual contribution estimated by Bayesian surprise, ICES directly assigns the exploration credits to individual agents. This approach bypasses the need for credit assignment of global intrinsic rewards and alleviates the noisy TV problem brought by stochastic environment transitions. Secondly, with the distinct algorithms and objectives to optimize exploration and exploitation policies, ICES retains the original MARL goal of maximizing extrinsic rewards while enjoying the benefit of cooperative exploration.

This paper has two main limitations. First, the proposed ICES requires additional policy networks, which introduce extra training complexity compared with QMIX or QPLEX. Reducing such complexity might be worth exploring when scaling ICES to cases with more agents. Second, this paper only considers one-step latent state transitions, which may be insufficient as exploration guidance in more complicated scenarios. For future work, we aim to incorporate time abstraction in ICES to further improve its applicability.

Acknowledgements

We thank the anonymous reviewers for their valuable feedback and suggestions.

Impact Statement

This paper presents work whose goal is to advance the field of Reinforcement Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., and Efros, A. A. Large-scale study of curiosity-driven learning. In *Proceedings of the International Conference on Learning Representations*, 2018.

Chen, X., Liu, X., Zhang, S., Ding, B., and Li, K. Goal consistency: An effective multi-agent cooperative method for multistage tasks. In Raedt, L. D. (ed.), *Proceedings of the 31st International Joint Conference on Artificial Intelligence*, pp. 172–178. ijcai.org, 2022.

Chitnis, R., Tulsiani, S., Gupta, S., and Gupta, A. Intrinsic motivation for encouraging synergistic behavior. In *Proceedings of the International Conference on Learning Representations*, 2019.

Du, Y., Han, L., Fang, M., Liu, J., Dai, T., and Tao, D. Liir:

Learning individual intrinsic reward in multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

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

Hu, J., Wang, S., Jiang, S., and Wang, W. Rethinking the implementation tricks and monotonicity constraint in cooperative multi-agent reinforcement learning. In *Proceedings of the 2nd Blogpost Track at International Conference on Learning Representations*, 2023.

Itti, L. and Baldi, P. Bayesian surprise attracts human attention. In *Advances in Neural Information Processing Systems*, pp. 547–554, 2005.

Jaques, N., Lazaridou, A., Hughes, E., Gulcehre, C., Ortega, P., Strouse, D., Leibo, J. Z., and De Freitas, N. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pp. 3040–3049. PMLR, 2019.

Jeon, J., Kim, W., Jung, W., and Sung, Y. Maser: Multi-agent reinforcement learning with subgoals generated from experience replay buffer. In *Proceedings of the International Conference on Machine Learning*, pp. 10041–10052. PMLR, 2022.

Jo, Y., Lee, S., Yeom, J., and Han, S. FoX: Formation-aware exploration in multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12985–12994, 2024.

Kim, W. and Sung, Y. An adaptive entropy-regularization framework for multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pp. 16829–16852. PMLR, 2023.

Kurach, K., Raichuk, A., Stańczyk, P., Zając, M., Bachem, O., Espeholt, L., Riquelme, C., Vincent, D., Michalski, M., Bousquet, O., et al. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 4501–4510, 2020.

Li, C., Wang, T., Wu, C., Zhao, Q., Yang, J., and Zhang, C. Celebrating diversity in shared multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Liu, B., Pu, Z., Pan, Y., Yi, J., Liang, Y., and Zhang, D. Lazy agents: a new perspective on solving sparse reward problem in multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pp. 21937–21950. PMLR, 2023a.

Liu, I.-J., Jain, U., Yeh, R. A., and Schwing, A. Cooperative exploration for multi-agent deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pp. 6826–6836. PMLR, 2021.

Liu, Z., Zhu, Y., and Chen, C. NA2Q: Neural attention additive model for interpretable multi-agent q-learning. In *Proceedings of the International Conference on Machine Learning*, pp. 22539–22558. PMLR, 2023b.

Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Pieter Abbeel, O., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Ma, Z., Wang, R., Li, F.-F., Bernstein, M., and Krishna, R. Elign: Expectation alignment as a multi-agent intrinsic reward. In *Advances in Neural Information Processing Systems*, volume 35, pp. 8304–8317, 2022.

Mahajan, A., Rashid, T., Samvelyan, M., and Whiteson, S. Maven: Multi-agent variational exploration. In *Advances in neural information processing systems*, volume 32, 2019.

Mazzaglia, P., Catal, O., Verbelen, T., and Dhoedt, B. Curiosity-driven exploration via latent bayesian surprise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7752–7760, 2022.

McKee, K. R., Gemp, I., McWilliams, B., Duèñez-Guzmán, E. A., Hughes, E., and Leibo, J. Z. Social diversity and social preferences in mixed-motive reinforcement learning. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 869–877, 2020.

Oliehoek, F. A. and Amato, C. *A concise introduction to decentralized POMDPs*. Springer, 2016.

Parker, J., Nunes, E., Godoy, J., and Gini, M. Exploiting spatial locality and heterogeneity of agents for search and rescue teamwork. *Journal of Field Robotics*, 33(7): 877–900, 2016.

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

Samvelyan, M., Rashid, T., Schroeder de Witt, C., Farquhar, G., Nardelli, N., Rudner, T. G., Hung, C.-M., Torr, P. H., Foerster, J., and Whiteson, S. The starcraft multi-agent challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2186–2188, 2019.

Schmidhuber, J. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

Seuken, S. and Zilberstein, S. Improved memory-bounded dynamic programming for decentralized pomdps. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, pp. 344–351, 2007.

Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Son, K., Kim, D., Kang, W. J., Hostallero, D. E., and Yi, Y. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pp. 5887–5896. PMLR, 2019.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., 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*, pp. 2085–2087, 2018.

Swamy, G., Reddy, S., Levine, S., and Dragan, A. D. Scaled autonomy: Enabling human operators to control robot fleets. In *Proceedings of the 2020 IEEE International Conference on Robotics and Automation*, pp. 5942–5948. IEEE, 2020.

Terry, J., Black, B., Grammel, N., Jayakumar, M., Hari, A., Sullivan, R., Santos, L. S., Dieffendahl, C., Horsch, C., Perez-Vicente, R., et al. Pettingzoo: Gym for multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 15032–15043, 2021.

Wang, J., Ren, Z., Liu, T., Yu, Y., and Zhang, C. Qplex: Duplex dueling multi-agent q-learning. In *Proceedings of the International Conference on Learning Representations*, 2020.

Wang, T., Wang, J., Wu, Y., and Zhang, C. Influence-based multi-agent exploration. In *Proceedings of the International Conference on Learning Representations*, 2019.

Xu, P., Zhang, J., and Huang, K. Exploration via joint policy diversity for sparse-reward multi-agent tasks. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*, pp. 326–334, 2023a.

Xu, P., Zhang, J., Yin, Q., Yu, C., Yang, Y., and Huang, K. Subspace-aware exploration for sparse-reward multi-agent tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11717–11725, 2023b.

Ying, W. and Dayong, S. Multi-agent framework for third party logistics in e-commerce. *Expert Systems with Applications*, 29(2):431–436, 2005.

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

Yuan, L., Zhang, Z., Li, L., Guan, C., and Yu, Y. A survey of progress on cooperative multi-agent reinforcement learning in open environment. *arXiv preprint arXiv:2312.01058*, 2023a.

Yuan, M., Li, B., Jin, X., and Zeng, W. Automatic intrinsic reward shaping for exploration in deep reinforcement learning. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 40531–40554. PMLR, 2023b.

Zhang, J., Zhang, Y., Zhang, X. S., Zang, Y., and Cheng, J. Intrinsic action tendency consistency for cooperative multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17600–17608, 2024.

Zhang, S., Cao, J., Yuan, L., Yu, Y., and Zhan, D.-C. Self-motivated multi-agent exploration. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pp. 476–484, 2023.

Zheng, L., Chen, J., Wang, J., He, J., Hu, Y., Chen, Y., Fan, C., Gao, Y., and Zhang, C. Episodic multi-agent reinforcement learning with curiosity-driven exploration. In *Advances in Neural Information Processing Systems*, volume 34, pp. 3757–3769, 2021.

Zhou, M., Liu, Z., Sui, P., Li, Y., and Chung, Y. Y. Learning implicit credit assignment for cooperative multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 11853–11864, 2020.

A. ICES Algorithm Details

A.1. ICES Exploration Policy Gradient

Here we detail the deviation to Equation (9). From Equation (8), we have:

$$\mathcal{J}_i(\xi) = \mathbb{E}_{\nu_i}[r^i_{\text{int}}] + \beta\mathcal{H}(\cdot|\tau^i, s). \tag{11}$$

Then, we introduce $V_\eta(\tau^i, s)$ as a baseline. Since $V_\eta(\tau^i, s)$ is independent to ν_i, we can rewrite the objective as:

$$\begin{aligned}\mathcal{J}_i(\xi) &= \mathbb{E}_{\nu_i(\xi)}\left[r^i_{\text{int}} - V_\eta(\tau^i, s)\right] + \beta\mathcal{H}(\cdot|\tau^i, s) &(12)\\ &= \mathbb{E}_{s,\tau^i\sim d,\nu_i(\xi)}\left[R^i_{\text{int}}(s,a) - V_\eta(\tau^i, s)\right] + \beta\mathcal{H}(\cdot|\tau^i, s), &(13)\end{aligned}$$

where d is the distribution of τ^i and s, and $R^i_{\text{int}}(\cdot,\cdot)$ is the intrinsic reward function.

Taking the gradient of $\mathcal{J}_i(\xi)$ with respect to ξ, we have:

$$\nabla_\xi\mathcal{J}_i(\xi) = \nabla_\xi\mathbb{E}_{s,\tau^i\sim d,\nu_i(\xi)}\left[R^i_{\text{int}}(s,a) - V_\eta(\tau^i, s)\right] + \beta\nabla_\xi\mathcal{H}(\cdot|\tau^i, s). \tag{14}$$

Define $r_{sa} = \mathbb{E}\left[R^i_{\text{int}}(s,a) - V_\eta(\tau^i, s)|s = s, a = a\right]$, we can rewrite the gradient as:

$$\begin{aligned}\nabla_\xi\mathcal{J}_i(\xi) &= \nabla_\xi\sum_{\tau^i,s} d(\tau^i, s)\sum_a \nu_{i,\xi}(a|s)r_{sa} + \beta\nabla_\xi\mathcal{H}(\cdot|\tau^i, s) &(15)\\ &= \sum_{\tau^i,s} d(\tau^i, s)\sum_a r_{sa}\nabla_\xi\nu_{i,\xi}(a|\tau^i, s) + \beta\nabla_\xi\mathcal{H}(\cdot|\tau^i, s) &(16)\\ &= \sum_{\tau^i,s} d(\tau^i, s)\sum_a r_{sa}\cdot\nu_{i,\xi}(a|\tau^i, s)\frac{\nabla_\xi\nu_{i,\xi}(a|\tau^i, s)}{\nu_{i,\xi}(a|\tau^i, s)} + \beta\nabla_\xi\mathcal{H}(\cdot|\tau^i, s) &(17)\\ &= \sum_{\tau^i,s} d(\tau^i, s)\sum_a \nu_{i,\xi}(a|\tau^i, s)\cdot r_{sa}\nabla_\xi\ln\nu_{i,\xi}(a|\tau^i, s) + \beta\nabla_\xi\mathcal{H}(\cdot|\tau^i, s) &(18)\\ &= \mathbb{E}_{s,\tau^i\sim d,\nu_i(\xi)}\left[\left(R^i_{\text{int}}(s,a) - V_\eta(\tau^i, s)\right)\cdot\ln\nu_{i,\xi}(a|\tau^i, s)\right] + \beta\nabla_\xi\mathcal{H}(\cdot|\tau^i, s) &(19)\\ &= \mathbb{E}_{s,\tau^i\sim d,\nu_i(\xi)}\left[\left(R^i_{\text{int}}(s,a) - V_\eta(\tau^i, s) - \beta\right)\cdot\ln\nu_{i,\xi}(a|\tau^i, s)\right]. &(20)\end{aligned}$$

A.2. ICES Training Details

We detail the training procedures for the policy networks and the scaffolds network mentioned in Section 3.4 with Algorithm 2 and Algorithm 3, respectively.

Algorithm 2 TrainPolicies: Training Procedure of ICES Policies (Section 3.3)

```
Input: Scaffolds parameters $\psi, \phi$, exploration network parameters $\xi, \eta$, exploitation networks parameters $\zeta$, replay buffer $\mathcal{D}$
Sample batch $\sim \mathcal{D}$
$\zeta \leftarrow \zeta - \text{LearningRate} \cdot \nabla\mathcal{L}(\zeta)$        {▷ Equation (7)}
for $i = 1, 2, ..., n$ do
    Calculate intrinsic scaffolds $r^i_{t,\text{int}} = D_{\text{KL}}\left[p_\psi(z_{t+1}|s_t, \boldsymbol{u}_t) \parallel p_\phi(z_{t+1}|s_t, \boldsymbol{u}_t^{-i})\right]$        {▷ Equation (2)}
end for
$\xi \leftarrow \xi + \text{LearningRate} \cdot \sum_i^n \nabla\mathcal{J}_i(\xi)$        {▷ Equation (8)}
$\eta \leftarrow \eta - \text{LearningRate} \cdot \nabla\mathcal{L}(\eta)$        {▷ Equation (10)}
Output: Updated parameters $\xi, \eta, \zeta$
```

B. Experiment Details

B.1. Environmental Settings

Google Research Football (GRF): We evaluate our proposed method ICES against baselines on three GRF (Kurach et al., 2020) scenarios, namely `academy_3_vs_1_with_keeper`, `academy_corner` and

Algorithm 3 TrainScaffolds: Training Procedure of ICES Scaffolds (Section 3.2)

```
Input: Scaffolds parameters $\psi, \phi, \theta$, replay buffer $\mathcal{D}$
Sample batch $\sim \mathcal{D}$
$\psi \leftarrow \psi + \text{LearningRate} \cdot \nabla \mathcal{J}_\psi(\psi, \phi, \theta)$
$\phi \leftarrow \phi + \text{LearningRate} \cdot \nabla \mathcal{J}_\phi(\psi, \phi, \theta)$
$\theta \leftarrow \theta + \text{LearningRate} \cdot \nabla \mathcal{J}_\theta(\psi, \phi, \theta)$
                                                        {▷ Equation (3)}
Output: Updated parameters $\psi, \phi, \theta$
```

`academy_counterattack_hard`. The sparse reward settings are used for ICES, baselines, and ablations, where the rewards are only observed when scoring or losing the game (Li et al., 2021). The details of the reward setting is given in Table 1. This reward structure calls for high levels of cooperation among agents and is further complicated by the stochastic nature of opponents' policies. For GRF tasks (Figures 4(a) to 4(c)), we plot the average scores (with 1 for wining, 0 for a tie and -1 for losing) of test episodes with respect to the training timesteps.

Table 1: GRF rewards.

Event	Reward
Our team scores	+100
Opponent team scores	-1
Our team or the ball returns to our half-court	-1

StarCraft Multi-agent Challenge (SMAC): We further assess our proposed method ICES on five SMAC (Samvelyan et al., 2019) scenarios with sparse reward settings following the previous works (Jeon et al., 2022; Kim & Sung, 2023). The rewards are only given upon the death of units (allies or enemies), and details are listed in Table 3. We use four easy tasks and one hard task for benchmark, including `3m`, `8m`, `2s3z`, `2s_vs_1sc` and `5m_vs_6m`, as specified in Table 2. For SMAC tasks (Figures 4(d) to 4(h)), we plot the average win rate of test episodes over training timesteps.

Table 2: SMAC challenges.

Task	Ally Units	Enemy Units	Type
3m	3 Marines	3 Marines	homogeneous, symmetric
8m	8 Marines	8 Marines	homogeneous, symmetric
2s3z	2 Stalkers & 3 Zealots	2 Stalkers & 3 Zealots	heterogeneous, symmetric
2s_vs_1sc	2 Stalkers	1 Spine Crawler	homogeneous, asymmetric
5m_vs_6m	5 Marines	6 Marines	heterogeneous, asymmetric

B.2. ICES Implementation Details

For both ICES and baselines, parameter sharing among agents is adopted to improve the sample efficiency as well as lower the model complexity. For ICES specifically, we remove the ϵ-greedy exploration strategy after the epsilon annealing time, since the exploration policy is already random in its nature.

For GRF, we implement ICES based on the code framework PyMARL (Samvelyan et al., 2019). For SMAC, we implement ICES based on the code framework PyMARL 2 (Hu et al., 2023). In this environment, before agents conduct any meaningful exploration, they tend to prolong the episode by escaping instead of attacking the enemies. Previous methods avoid this behavior by normalizing the intrinsic rewards to be less than or equal to zero (Jeon et al., 2022; Jo et al., 2024), while we simply add a -0.02 step penalty as intrinsic rewards. For both GRF and SMAC experiments, default hyperparameters from the code frameworks are used. For ICES-specific hyperparameters, we list them in Table 4. In particular, α anneals gradually throughout the training process.

Table 3: SMAC rewards.

Event	Reward
All enemies die	+200
One enemy dies	+10
One ally dies	-5

Table 4: ICES hyperparameters.

Hyperparameter	Benchmark	Scenario	Value
Action embedding dimension	-	-	4
Scaffolds learning rate	-	-	0.0001
Scaffolds gradient clipping	-	-	0.1
Exploration agent learning rate	GRF	-	0.001
	SMAC	-	0.01
α	GRF	academy_3_vs_1_with_keeper	0.2 - 0.05
		academy_corner	0.2 - 0.05
		academy_counterattack_hard	0.1 - 0.05
	SMAC	5m_vs_6m	0.1 - 0.05
		others	0.1 - 0.1
β	GRF	academy_3_vs_1_with_keeper	0.02
		academy_corner	0.05
		academy_counterattack_hard	0.05
	SMAC	5m_vs_6m	0.5
		others	0.1

B.3. Infrastructure

Experiments are carried out on NVIDIA GeForce RTX 3080 GPUs.

C. Additional Experimental Results

C.1. ICES on KAZ

Since ICES do not require particular parsing of states, it can be easily generalized to pixel-based MARL tasks. Therefore, we further test ICES on a pixel-based MARL benchmark task `knights_archers_zombies` (KAZ) from the pettingzoo environments (Terry et al., 2021) with the results shown in Figure 10. We can see that in pixel-based MARL tasks, ICES is also able to improve the performance by promoting cooperative exploration.

Knights Archers Zombies (KAZ) Environment: In this game, we control 4 agents (2 knights and 2 archers) with the goal to kill all zombies that appear on the screen. Each agent can move and attack to kill zombies. When a knight attacks, it swings a mace in an arc in front of its current heading direction. When an archer attacks, it fires an arrow in a straight line in the direction of the archer's heading. We reward the agents with $+1$ when a zombie dies.

Inputs Preprocessing: We use the pixel-based local and global obaservation in this environment for ICES and QMIX. In particular, the global observation is represented by a $720 \times 1280 \times 3$ pixel colored image, while the observation of each agent is represented as a $512 \times 512 \times 3$ pixel colored image around the agent. As the input space is too large for RL algorithms to learn efficiently, we adopt some necessary preprocessing to local and global observation to reduce the dimension of input space.

The preprocessing pipeline for local observation is as follows: Color Reduction (Only take the first color channel and discard

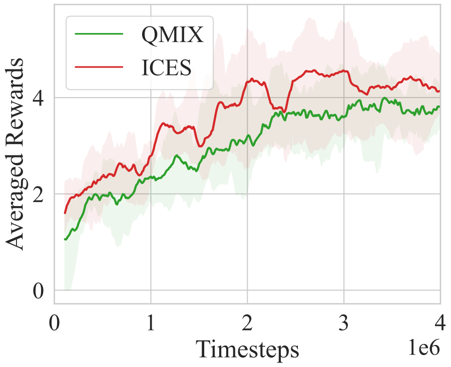


Figure 10: Performance comparison on KAZ benchmark.

the rest) → Resize to 64×64. The preprocessing pipeline for the global observation is as follows: Color Reduction → Crop the central 720×1100 pixels → Resize to 128×128.

Network Architecture: We use the code framework PyMARL 2 (Hu et al., 2023) for this experiment. Since the inputs are images rather than vectors, we add feature encoding blocks in the agent network and the mixing hypernetwork to process the input images. Details are provided in Table 5 and Table 6, respectively. RNN layers are not used for this experiment to avoid extensive GPU memory consumption.

Table 5: Observation encoding blocks in agent network.

Layer	Operator	# Channels
1	Conv 3×3 & MaxPooling	32
2	Conv 3×3 & MaxPooling	64
3	Conv 3×3 & MaxPooling	128
4	Flatten & FC	128

Table 6: Observation encoding blocks in mixing hypernetwork.

Layer	Operator	# Channels
1	Conv 5×5 & MaxPooling	32
2	Conv 5×5 & MaxPooling	64
3	Conv 5×5 & MaxPooling	128
4	Flatten & FC	128

For ICES, we use the same QMIX network for value functions and the CVAEs with network structure specified in Table 7 and Table 8. We design the network architecture for CVAE here based on `https://github.com/AntixK/PyTorch-VAE`.

Hyperparameters: We use the default hyperparameters in PyMARL 2 except for batch_size $= 4$ for both QMIX and ICES. For ICES specific hyperparameters, we list them in Table 9.

Table 7: CVAE encoder in ICES.

Layer	Operator	# Channels
1	Conv 5×5	8
2	Conv 5×5	16
3	Conv 5×5	32
4	Conv 3×3	64
5	Conv 3×3	128
6	Flatten & FC	128
7	FC $\times 2$	64

Table 8: CVAE decoder in ICES.

Layer	Operator	# Channels
1	FC $\times 2$	64
2	TransposedConv 5×5	64
3	TransposedConv 5×5	32
4	TransposedConv 5×5	16
5	TransposedConv 5×5	8
6	TransposedConv 7×7	2

Table 9: ICES hyperparameters used in KAZ experiment.

Hyperparameter	Value
Action embedding dimension	4
Scaffolds learning rate	0.0001
Scaffolds gradient clipping	0.1
Exploration agent learning rate	0.001
α	0.1 - 0.05
β	0.1