Improving Cooperative Multi-Agent Exploration via Surprise Minimization and Social Influence Maximization

Mingyang Sun
Dalian University of Technology
Dalian, China
mysun@mail.dlut.edu.cn

Yaqing Hou
Dalian University of Technology
Dalian, China
houyq@dlut.edu.cn

Jie Kang
Dalian University of Technology
Dalian, China
kangj@mail.dlut.edu.cn

Yifeng Zeng
Northumbria University
Newcastle upon Tyne, United Kingdom
yifeng.zeng@northumbria.ac.uk

Qiang Zhang
Dalian University of Technology
Dalian, China
zhangq@dlut.edu.cn

ABSTRACT

In multi-agent reinforcement learning (MARL), the uncertainty of state change and the inconsistency between agents' local observation and global information are always the main obstacles of cooperative multi-agent exploration. To address these challenges, in this paper, we propose a novel MARL exploration method by combining surprise minimization and social influence maximization. Considering state entropy as a measure of surprise, surprise minimization is achieved by rewarding the individual's intrinsic motivation (or rewards) for coping with more stable and familiar situations, hence promoting the policy learning. Furthermore, we introduce mutual information between agents' actions as a regularizer to maximize the social influence via optimizing a tractable variational estimation. In this way, the agents are guided to interact positively with one another by navigating between states that favor cooperation. We further empirically demonstrate the significant performance of the proposed exploration method in improving the cooperative ability of agents in a well known Multi-Agent MuJoCo environment.

KEYWORDS

Multi-Agent Reinforcement Learning, Exploration, Cooperative Multi-Agent

ACM Reference Format:
Mingyang Sun, Yaqing Hou, Jie Kang, Yifeng Zeng, and Qiang Zhang. 2023. Improving Cooperative Multi-Agent Exploration via Surprise Minimization and Social Influence Maximization. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 7 pages.

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) has to date achieved excellent success due to its ubiquity of a wide realm of real world problem domains including crewless aerial vehicles [21], automatic traffic light control [4], and cooperative robot control [16], etc. Previous studies on MARL tend to apply single-agent RL algorithms in multi-agent scenarios and promote independent learning [4] [6]. However, the algorithms often suffer from non-stationarity during the learning as they simply treat other learning agents as part of an environment. To address this issue, recent work has introduced a centralized training and distributed execution (CTDE) paradigm and more advanced techniques such as multi-agent deep deterministic policy gradient (MADDPG) [12], monotonic value function factorization (QMIX) [17], Multi-Agent Proximal Policy Optimization (MAPPO) [23] and Heterogeneous-Agent Proximal Policy Optimisation (HAPPO) [10] have been developed. Nevertheless, despite their excellent performance as reported, the aforementioned methods still adopt the classical noise-based exploration strategies following the single-agent RL methods as a noisy version of the actor policy [11], e.g., ϵ-greedy exploration in QMIX and entropy regularization in MAPPO, hence falling short of taking into account agents' interactions in environment exploration and resulting in slow exploration and sub-optimality [22].

More recent studies have been proposed to solve the challenge of efficiently exploring unknown environments and gleaning informative experiences that could benefit the policy learning most towards optimal ones. For example, MAVEN was proposed to improve exploration by conditioning an agent's behavior on a shared latent variable controlled by a hierarchical policy [13]. Further, Wang [20] encouraged coordinated exploration by considering the influence of one agent's behavior on other agents' behaviors. However, while these exploration strategies for MARL obtain promising learning performance, they suffer from two common issues: (*a*) the partial observation and non-stationary problems induce extra difficulty in the exploration measurement. While individual agents act according to their own policies under local observations, they do not share the same knowledge of environmental states. Furthermore, the behaviors and states of individuals can also be influenced by their counterparts, which induces additional randomness in the learning; and (*b*) Even in coordinated exploration, the inconsistency between local and global information may exist. This demands agents to balance the learning from both local and global perspective; otherwise, it may lead to inadequate or redundant exploration. Thus, how to explore effectively in more general scenarios with information bias and cooperative dependency among multiple agents remains an open and challenging research problem.

This paper attempts to take a step towards solving the above issues. First, we observe that uncertainties in the environment can

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

keep an agent in an unstable state of change, which is not conducive to exploration and learning. An example was illustrated in [3], where the environment around an agent is unstable due to weather changes, and if it builds a shelter and hides in it, although this behavior leads the agent to go through some unfamiliar states initially, it can reach a stable and predictable state in the long run. Conversely, if it does not take such measures, then it will constantly experience unstable states. Therefore, we believe that explicitly preventing the agents from exploring states with a high degree of arbitrary uncertainty is an important prerequisite for improving the efficiency and robustness of exploration. To do this, we introduce *surprise minimization* to cope with unpredictable state changes in multi-agent scenarios. However, if only surprise is minimized, it is easy for the agents to adopt negative or conservative policies, which is detrimental to their learning of cooperative behavior. Cooperative behaviors usually emerge because there is a lot of interaction between the cooperators. The decisions of one of the cooperators are influenced by the behaviors from the other parties, thus facilitating the emergence of effective cooperation. In multi-agent reinforcement learning, the influence of an agent's behavior on the decisions of other individuals is called social influence [7]. *Social Influence Maximization* of policies encourages agents to interact actively with each other rather than blindly performing unusual actions, thus enabling them to learn to cooperate effectively. Therefore, we believe that it is worthwhile to use global information to maximize social influence and thus promote more effective interactions between agents. In summary, this paper proposes a new multi-agent exploration approach with surprise minimization and social influence maximization, where surprise minimization and social influence maximization complement each other and jointly promote cooperative behavior. A more stable state distribution reduces the range of states involved in maximizing social influence. Conversely, social influential exploration encourages agents to navigate between the states that favor cooperation.

In general, we make the following main contributions:

- We propose a novel MARL exploration method by combining *surprise minimization* and *social influence maximization*, named S2MIA. This new method reduces the impact of state uncertainty on agent learning and encourages agents to interact for the emergence of cooperative behaviors.
- We consider state entropy as a measure of surprise. To achieve surprise minimization, state probabilities in trajectories are maximized by rewarding the individual's intrinsic motivation (rewards) for learning each agent's policy. Furthermore, we introduce mutual information between agents' actions as a regularizer to maximize the social influence via optimizing a tractable variational estimation.
- We apply the proposed method to the advanced HAPPO [10] algorithm and present the implementation in details. We demonstrate the effectiveness of S2MIA on the Multi-agent Mujoco environment. Empirical results show S2MIA can improve the learning by guiding and coordinating agents' exploration.

The rest of the paper is organized as follows. In Section 2, we describe the problem formulation, as well as backgrounds of surprise minimization and exploration in multi-agent reinforcement learning.

2 PROBLEM FORMULATION AND BACKGROUNDS

2.1 Surprise Minimization

Surprise can be inferred as a measure of deviation among states encountered by the agent during its interaction with the environment [3, 5]. In stable or simplified RL environment, surprise driven exploration methods have been proposed and mostly focused on pursuing surprises or satisfying curiosity by means of state visitation counting, measuring prediction error, maximizing action entropy or state entropy and so on, aiming to obtain richer environmental information [22]. However, these methods tend to suffer from the issue of inefficient sampling [1], particularly in MARL scenarios with a larger number of agents. On the other hand, surprise minimization is the exact opposite - it expects the agent to be in a stable state space, hence can understand its surroundings more efficiently. In model-based reinforcement learning, surprise minimization helps agents model policies by speeding up the convergence but easily get trapped in local optimum as the agents tend to make decisions with conservative actions [8]. The surprise minimization is witnessed as an intrinsic motivation [1] or lifting generalization problem [5]. [19] uses the free energy to estimate and minimize surprise jointly across all agents, which is combined with value decomposition method to explore the promotion effect of surprise minimization on multi-agent learning. Taking this clue, this paper introduces surprise minimization to cope with the state uncertainty or unpredictable state changes in multi-agent scenarios. Besides, the social influence maximization is further proposed to encourage the cooperative exploration in the complex multi-agent learning environment.

2.2 Exploration in MARL

Compared to the success in a single-agent domain, the study on exploration for MARL is still at the preliminary stage. For example, from the perspective of uncertainty-oriented exploration, Zhu [24] proposed Multi-agent Safe Q-Learning using the epistemic uncertainty for exploration in accordance with the optimism principle. Martin and Sandhom [14] used both the epistemic uncertainty and aleatoric uncertainty following the similar idea of distributional value estimation in single-agent RL [2]. These methods only introduce uncertainty for better state-action value estimation and rarely consider how to balance local and global information to obtain a robust and accurate uncertainty estimation particularly when local information is inconsistent with global information.

Intrinsic motivation-oriented exploration is also a promising direction and is gradually tending to be applied in a multi-agent domain. Strouse [18]took mutual information of the goal and the agents' states or actions in the form of internal rewards as a means to help the agent learn to hide or share intentions. Wang [20] measured influence of one agent on other agents' transition function (EITI) and rewarding structure (EDTI). These two measures encouraged agents to visit critical states in the state-action space,

through which agents can transit to potentially important under-explored regions. Closely related to our work, Jagues [7] also defined the intrinsic reward function by introducing counterfactual reasoning from the perspective of social influence. By maximizing this function, agents are encouraged to take actions with the most strong influence on the policies of other agents through causal inference or communication. However, circular dependencies and limited communication will be the main obstacles to the generalization of this method to practical applications. In our work, the variational posterior estimator built by extending the policy network does not have these limitations.

3 OUR METHODS

3.1 Preliminaries

We formulate the fully cooperative multi-agent task as a Dec-POMDP, which is formally defined as a tuple $G = \langle S, A, P, r, \Omega, O, n, \gamma \rangle$. Here, S is the finite state space of the environment. At each time step t, every agent $i \in N \equiv \{1, \ldots, n\}$ chooses an action $a^i \in A$ which forms the joint action $\mathbf{a} \in \mathbf{A} \equiv A^n$. $P(s'|s, \mathbf{a}) : S \times \mathbf{A} \times S \to [0, 1]$ is the state transition function. $r(s; \mathbf{a}) : S \times \mathbf{A} \to \mathbb{R}$ is the reward function shared by all agents and $\gamma \in [0, 1)$ is the discount factor. We consider *partially observable* settings, where an agent only has access to an observation $o_i \in \Omega$ drawn according to observation function $O(s, i) : S \times N \to \Omega$, not its true state s^i. The action-observation history for an agent i is $\tau^i \in T \equiv (\Omega \times A)^*$ on which it can condition its policy $\pi^i(a^i|\tau^i) : T \times A \to [0, 1]$. We use $\boldsymbol{a}^{-i}$ to denote the action of all the agents other than i and follow a similar convention for the policies π^{-i}. For a joint policy $\boldsymbol{\pi}$, the state-value function and the action-value function are defined: $V_{\boldsymbol{\pi}}(s_t) = \mathbb{E}_{\mathbf{a}_{0:\infty} \sim \boldsymbol{\pi}, s_{1:\infty} \sim P}[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t]$, $Q^{\pi}(s_t; \mathbf{a}_t) = \mathbb{E}_{s_{t+1:\infty}, \mathbf{a}_{t+1:\infty}}[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t, \mathbf{a}_t]$. The goal of the problem is to find the optimal joint policy $\boldsymbol{\pi}^*$ by maximize the expected total reward:

$$\mathcal{J}(\boldsymbol{\pi}) \triangleq \mathbb{E}_{\rho_0, \boldsymbol{\pi}, P}[\sum_{t=0}^{\infty} \gamma^t r_t], \tag{1}$$

where ρ_0 represents the initial state distribution, and $s_0 \sim \rho_0(s_0)$, $s_{t+1} \sim P(\cdot|s_t, a_t)$, $\mathbf{a}_t \sim \boldsymbol{\pi}(s_t)$.

We use the centralized training and Decentralized Execution (CTDE) paradigm, which has recently been widely adopted by deepMARL, to design effective exploration method for agents.

3.2 Individual Surprise Minimization

The unpredictability of the environment means high entropy of state, which directly leads to surprises. In the decentralized execution mode, especially in partially observable cases, from the perspective of a single agent, the long term effects of all agents' actions on its surprise are complex. The change of an agent's own state is determined by its own actions and the actions of other agents. Moreover, the policies from other agents also need to be taken into account when it is constructing its beliefs about the future. But the policies of each agent are independent and often unknown from each other. In the case of limited communication, an agent cannot accurately know the next action of others, and it is even more difficult to directly intervene in the decisions of others. Therefore, surprise is easy to come by when agents behave in unexpected ways, such as blind exploration. The surprises caused by this series of reasons make it difficult for the agent to evaluate whether its behavior is really meaningful.

In this section, we will present our deep RL algorithm for learning decentralized policies that minimize surprise. Considering the huge global state space and joint action space, it is difficult to treat global surprise minimization as a centralized goal. Meanwhile, it is not entirely appropriate as agents' actions have different degrees of impact on the global surprise. Thus, we propose surprise minimization as the individual intrinsic motivation by seeking out low entropy state distributions.

3.2.1 Surprise Minimization as Intrinsic Rewards. To incorporate individual surprise minimization into agents' learning, we design intrinsic rewards as an embodiment of the extra benefit an agent gets when it experiences more familiar states, based on the history of the states it experiences under the current policy. We assume an agent i learns a policy π^i_ϕ, parameterized by ϕ. The goal of surprise minimization is to minimize the entropy of its state marginal distribution under its current policy π^i_ϕ at each time step of the episode. This state entropy can be estimated by fitting an estimate of the visited state marginal $p^{\pi^i_\phi}(s^i_t)$ as each step t, given by $p_{\theta^i}(s^i_t)$, using the states during the entire episode. For a complete trajectory $\tau^i = \{s^i_1, \ldots, s^i_t \ldots, s^i_T\}$, We can get an upper bound as an estimate of the sum of the entropy of each state distribution over the whole episode:

$$\begin{aligned} \sum_{t=0}^{T} \mathcal{H}(s^i_t) &= -\sum_{t=0}^{T} \mathbb{E}_{s^i_t \sim p^{\pi^i_\phi}(s^i_t)}[\log p^{\pi^i_\phi}(s^i_t)] \\ &\leq -\sum_{t=0}^{T} \mathbb{E}_{s^i_t \sim p^{\pi^i_\phi}(s^i_t)}[\log p_{\theta^i}(s^i_t)], \end{aligned} \tag{2}$$

where the inequality becomes an equality when $p_{\theta^i}(s^i_t)$ accurately models $p^{\pi^i_\phi}(s^i_t)$. Minimizing the right-hand side of the inequality is equivalent to a new reinforcement learning maximization objective with an additional internal reward. This leads to a new reward function:

$$\tilde{r}(s^i_t) = r(s^i_t) + \alpha \log p_{\theta^i}(s^i_t), \tag{3}$$

where the coefficient α is used to control the proportion of intrinsic reward. The most basic principle is to put these two reward terms on a similar magnitude.

3.2.2 Density Estimation. In order to instantiate surprise minimization into MARL algorithms, the sufficient statistics of $p^i_\theta(s^i)$ in Equation 2 is essential. Considering that the state marginal distribution $p^{\pi^i_\phi}(s^i_t)$ changes with the update of parameters of π^i_ϕ , the distribution $p^i_\theta(s^i)$ also needs to update accordingly. Specifically, at the k^{th} iteration, all agents obtain new policies $\boldsymbol{\pi}_{\phi_k}$ by a policy gradient algorithm. Next, all agents interact with the environment according to the policy $\boldsymbol{\pi}_{\phi_k}$ and collect trajectories data to update into the buffer D_k. Before the next iteration $k+1$ begins, for each agent, the parameters of the sufficient statistics $\theta^i_k = \mathcal{U}(D_k)$ are first recalculated using a maximum likelihood state density estimation process $\theta^i_k = \arg\max_{\theta^i} \sum_{D_k} \log p_{\theta^i}(s^i)$ over the experience within the trajectory buffer D_k.

In principle, any appropriate model class can be selected according to the training environment to estimate the density $p_\theta(s^i)$. Relatively simple distribution classes, such as products of independent marginals, suffice to run our methods in many environments. As we show in our experiments (see Section 4), $p_\theta(s^i)$ is simply modeled as an independent Gaussian distribution for each dimension of the observation. Thus, the formula for the full reward can be rewritten as:

$$\tilde{r}(s^i) = r(s^i_j) - \alpha \sum_j (\log \sigma_j + \frac{(s^i_j - \mu_j)^2}{2\sigma_j^2}), \tag{4}$$

where μ_j and σ_j are separately calculated as the sample mean and standard deviation at j^{th} dimension of the state space from trajectory buffer.

However, in more complex environments, when the state of an agent or the dimensionality of observations (such as images) is large, it is advisable to employ a more sophisticated density estimator or utilize some dimensionality reduction and feature extraction techniques (such as Variational Auto-Encoders [9]).

3.3 Social Influential Exploration

Social influence measures the influence of one agent's action on others' behavior. Actions that lead to relatively higher change in the other agent's behavior are considered to be highly influential. Social influence maximization is related to maximizing the mutual information (MI) between agents' actions, and hypothesize that this inductive bias will drive agents to learn coordinated behavior.

In this section, we introduce social influence as a regularization term, which stimulates agents to maximize the mutual information between their actions. This regularization term $I(i, -i)$ can be added the objective function of policy optimization algorithm:

$$\tilde{\mathcal{J}} = \mathcal{J}(\pi^i) + \lambda I(i, -i), \tag{5}$$

where λ is used to control the intensity of the social influential regularization term.

By sampling sufficient joint actions, and averaging the resulting policy distribution of i in each case, we can obtain the marginal policy of i, $p(a^i_t|s^i_t) = \sum_{\boldsymbol{a}^{-i}_t} p(a^i_t|\boldsymbol{a}^{-i}_t, s^i_t)p(\boldsymbol{a}^{-i}_t|s^i_t)$, that is the decentralized policy of i without considering the actions of other agents. The discrepancy between the marginal policy of i and the conditional policy of i given $-i$'s action is a measure of the causal influence of $-i$ on i; it gives the degree to which i changes its planned action distribution because of actions of other agents. The influence regularization term to the mutual information between the actions of agents i and $-i$, which is given by

$$\begin{aligned} I(A^i; A^{-i}|s^i) &= \sum_{\mathbf{a}} p(\mathbf{a}|s^i) \log \frac{p(\mathbf{a}|s^i)}{p(a^i|s^i)p(\boldsymbol{a}^{-i}|s^i)} \\ &= \sum_{\boldsymbol{a}^{-i}} p(\boldsymbol{a}^{-i}|s^i) D_{KL}[p(a^i|\boldsymbol{a}^{-i}, s^i) \| p(a^i|s^i)]. \end{aligned} \tag{6}$$

To maximize the social influence, estimating $p(a^i|\boldsymbol{a}^{-i}, s^i)$ is the main obstacle. By sampling N independent trajectories from the environment, we can perform a Monte-Carlo (MC) approximation

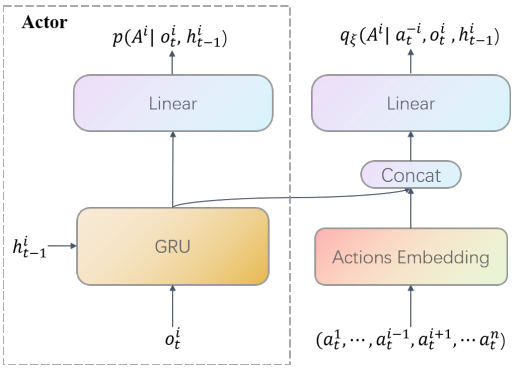


Figure 1: The architecture of policy network. The left is agent i's Actor network, which receives the current local observations o^i_t and the last hidden states h^i_{t-1}. The right is the variational posterior estimator about $q_\xi(a^i|\boldsymbol{a}^{-i}, s^i)$ by building an actions encoder.

of the MI:

$$\begin{aligned} I(A^i; A^{-i}|s^i) &= \mathbb{E}_\tau[D_{KL}[p(a^i|\boldsymbol{a}^{-i}, s^i) \| p(a^i|s^i)]] \\ &\approx \frac{1}{N} \sum_n D_{KL}[p(a^i|\boldsymbol{a}^{-i}_n, s^i) \| p(a^i|s^i)]. \end{aligned} \tag{7}$$

If the state and action spaces are small, we simply count the frequencies $N_1(a^i, \boldsymbol{a}^{-i}, s^i)$ and $N_2(\boldsymbol{a}^{-i}, s^i)$ of tuples $(a^i, \boldsymbol{a}^{-i}, s^i)$ and $(\boldsymbol{a}^{-i}, s^i)$ separately from the samples, and then calculate their ratios $\frac{N_1(a^i, \boldsymbol{a}^{-i}, s^i)}{N_2(\boldsymbol{a}^{-i}, s^i)}$ as an accurate estimate of $p(a^i|\boldsymbol{a}^{-i}, s^i)$.

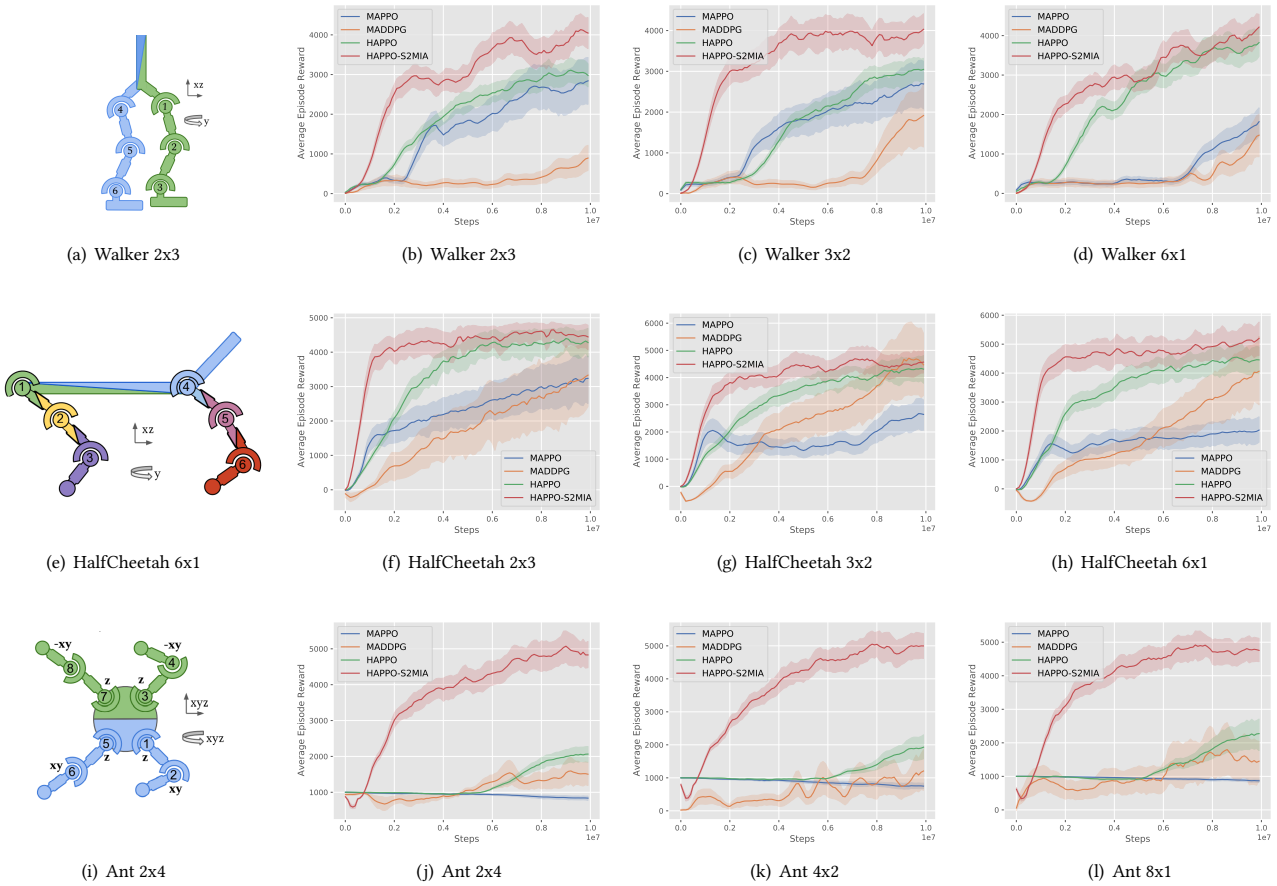
Unfortunately, in many multi-agent scenarios, where the problem space is often large, the amount of memory consumed by MC is often unrealistic for accurate estimation. As an alternative, for the mutual information objective, we introduce a variational posterior $q_\xi(a^i|\boldsymbol{a}^{-i}, s^i)$ via a neural network with parameters ξ to derive a tractable lower bound:

$$I(A^i; A^{-i}|s^i) \geqq \sum_{\boldsymbol{a}^{-i}} p(\boldsymbol{a}^{-i}|s^i) D_{KL}[q_\xi(a^i|\boldsymbol{a}^{-i}, s^i) \| p(a^i|s^i)]. \tag{8}$$

3.4 Implementation Details

In the present study, we incorporate surprise minimization and social influential regularization into a state-of-the-art MARL algorithm, namely HAPPO [10], as an easy-to-apply exploration technique. Different from the majority of existing MARL methods, i.e., QMIX and MAPPO, HAPPO was recently proposed as a multi-agent trust region method and is immune from restriction under parameter sharing and homogeneity of multiple agents which can lead to a sub-optimal outcome that is exponentially worse with a larger number of agents [10].

As our interest is placed on solving Decentralized Partially Observable Markov Decision Processes with Shared Rewards (DEC-POMDP), different agents merely have access to their local observations. For simplicity, we use agents' local observations instead of fully observable states in intrinsic rewards maximization of Equation 3. Accordingly, the reward function with surprise minimization



(a) Walker 2x3 (b) Walker 2x3 (c) Walker 3x2 (d) Walker 6x1

(e) HalfCheetah 6x1 (f) HalfCheetah 2x3 (g) HalfCheetah 3x2 (h) HalfCheetah 6x1

(i) Ant 2x4 (j) Ant 2x4 (k) Ant 4x2 (l) Ant 8x1

Figure 2: (left) Agent partitionings for Multi-Agent MuJoCo environments: Walker [2x3], HalfCheetah [6x1] and Ant [2x4]. Colours indicate agent partitionings. The number is the ID of the controlled joints. (right) Mean performance for multiple Multi-Agent MuJoCo tasks over 10 runs with a fixed set of seeds, with interquartile ranges shown in shaded areas. The x-axis is the number of environmental steps during training; The y-axis is average episode rewards during testing. HAPPO-S2MIA outperforms its rivals.

as the intrinsic motivation is then re-formalized as

$$\tilde{r}(\mathbf{s}_t, o_t^i) = r(\mathbf{s}_t) + \alpha \log p_{\theta^i}(o_t^i). \tag{9}$$

Then, $\tilde{r}$ is directly used to estimate the advantage of actions following common practices in HAPPO implementations.

Furthermore, as shown in Fig. 1 (left), each agent is assigned with a policy network as Actor, which uses a GRU as trajectory encoder to encode the agent's observation o_t^i and internal state h_{t-1}^i. After introducing the social influential regularization, we augment the initial policy network with an addition policy output head conditioned a joint action encoder for the agents to estimate q_ξ in Equation 8. During centralized training, the whole policy network will use the joint action encoder to encode the actions of other agents $\boldsymbol{a}_t^{-i}$, and then through the two output headers to calculate $p(a_t^i|s_t^i) \approx p(a_t^i|o_t^i, h_{t-1}^i)$ and $q_\xi(a_t^i|\boldsymbol{a}_t^{-i}, s_t^i) \approx q_\xi(a_t^i|\boldsymbol{a}_t^{-i}, o_t^i, h_{t-1}^i)$ simultaneously. Finally, the policy network is trained to maximize the objective

$$\begin{aligned}\tilde{L}(\theta) &= L_{HAPPO}(\theta) \\ &+ \lambda \frac{1}{BT} \sum_{b=1}^{B} \sum_{t=0}^{T} D_{KL}[q_\xi(\cdot|\boldsymbol{a}^{-i}, s^i) \| p(\cdot|s^i)],\end{aligned} \tag{10}$$

where B is the size of mini-batch and L_{HAPPO} represents the original optimization objective of HAPPO (details can be seen from [10]).

4 EXPERIMENTAL STUDY

In this section, we conducted a series of experiments to evaluate the theoretical claims presented by S2MIA along with its performance.

4.1 Experimental Setup

To verify the adaptability of S2MIA, we apply it on Heterogeneous-Agent Proximal Policy Optimisation (HAPPO) as described in Section 3.4. We benchmark HAPPO-S2MIA against other existing state-of-the-art (SOTA) algorithms, which include MAPPO, MADDPG and the original HAPPO. Parameter Settings of MAPPO, HAPPO and MADDPG, such as learning rate, neural network structure and ppo-clip rate are consistent with previous works. S2MIA additionally introduces two important parameters, i.e., intrinsic reward coefficient α and regular term coefficient λ. To make the environmental reward and intrinsic reward orders of magnitude close, α is set to 0.01. We simply replace HAPPO's original entropy term with the social influence regularizer, so the coefficient remains unchanged at 0.01.

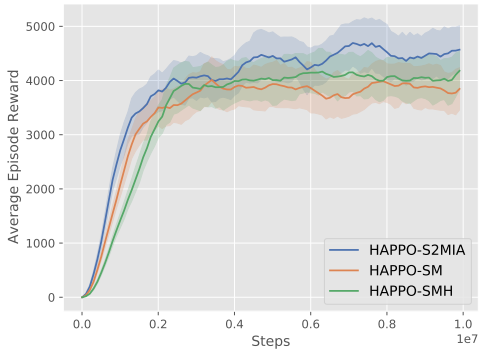
We choose the Multi-Agent MuJoCo [15] as the testbed for its rich environments and high complexity of control. MuJoCo tasks challenge a robot to learn an optimal control policy of motion and Multi-Agent MuJoCo assigns control of each part of the robot to multiple independent agents. With the increasing variety of the body parts, it becomes necessary to model heterogeneous policies, such as in the HalfCheetah [6x1] task shown in Fig. 2(e), where different agents control different types of joints. This fits perfectly with our method, as one individual's intrinsic rewards should not be involved in the updating of other individuals' policies.

4.2 Results

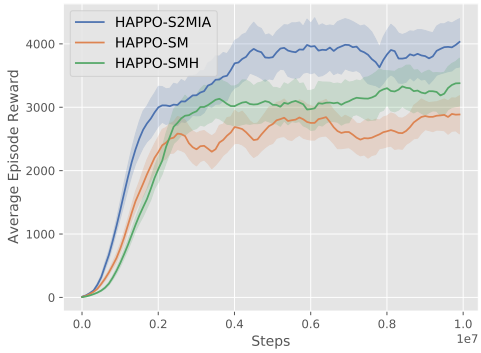
We consider a total of 9 tasks in 3 different scenarios, i.e., Walker, HalfCheetah and Ant, in Multi-Agent MuJoCo to conduct our experiments. The results are shown in Fig. 2. Agents were evaluated for a total of 10 million environmental steps with the lines in the plot indicating average episode rewards and the shaded area as 95% confidence interval over 10 independent runs. The plots show that HAPPO-S2MIA performs substantially better than all rivals on every control task. Specifically, although HAPPO, as the most advanced algorithm among them, achieved a significant advantage in most tasks, the improvement of average episode reward in the early stage of learning was not satisfactory. For example, in 3 Ant tasks, we can only observe the obvious learning effect after about 6 million environmental steps. By comparison, the improvement brought by our exploration method S2MIA to HAPPO is particularly significant. A significant improvement in early learning efficiency was observed in all tasks. This suggests that under the combined effects of surprise minimization and social influence maximization, agents can discover policies that keep the task going more quickly. The final test result of HAPPO-S2MIA is better than HAPPO, which also shows that our method can explore better cooperation policies than simple entropy regularization.

4.3 Ablation Study

To better understand the approach, we perform an ablation study to verify the effectiveness of social influencial exploration. At the same time, we test the advantage of social influence regularization over HAPPO's original entropy regularization. For clarity, the HAPPO using surprise minimization without exploration is called HAPPO-SM, while based on HAPPO-SM, the algorithm with the original entropy regularization is called HAPPO-SMH. As Fig. 3 shows, although HAPPO can obtain a satisfactory reward growth in the early stage by introducing surprise minimization, if there is no exploration, surprise minimization will limit it to learn sub-optimal policies. After the introduction of entropy regularization (i.e., HAPPO-SMH), this problem is alleviated to a certain extent, such as an increase of about 600 average episode rewards in the Walker task. In addition, if we use our social influence regularization to replace entropy regularization, the experimental performance will be significantly improved. For example, in the same Walker task, HAPPO-S2MIA received about 1000 reward increases compared to HAPPO-SM, and obtained competitive the early learning performance compared to HAPPO-SMH. This demonstrates that the interaction between surprise minimization and social influence maximization can effectively help agents explore better cooperation policies.



(a) HalfCheetah 3x2



(b) Walker 3x2

Figure 3: Comparison of our methods against ablations for HalfCheetah and Walker. HAPPO-SM refers to the HAPPO algorithm using surprise minimization but without any means of exploration, while HAPPO-SMH uses the entropy regularization originally used by HAPPO.

5 CONCLUSION

This paper presents a novel multi-agent exploration method with *surprise minimization* and *social influence maximization* for addressing the specific challenges of state uncertainty that arise in cooperative multi-agent learning. Superficially, in order to minimize surprise, a distribution of experienced states is constructed and the probability of each state is considered an intrinsic reward. This intrinsic reward makes an agent's policy more likely to pursue low state entropy. The surprise minimization is regarded as the internal motivation of individuals, while the maximization of social influence is regarded as the common goal across individuals. We use mutual information between agent actions to measure social influence and add it as a regularizer to the objective function. And we design a variational posterior estimator for computing this regularizer. We evaluated our method over nine complex multi-agent control tasks on Multi-Agent MuJoCo, and the results show that the proposed methods increases exploration efficiency.

Although we only apply the proposed method to the HAPPO algorithm, this exploration method can easily be generalized to many other MARL algorithms. This work complements current research on cooperative multi-agent exploration and provides effective solutions, yet how to deal with the exponential growth of global state and action spaces are still a open problem.

REFERENCES

[1] Joshua Achiam and Shankar Sastry. 2017. Surprise-based intrinsic motivation for deep reinforcement learning. *arXiv preprint arXiv:1703.01732* (2017).

[2] Marc G Bellemare, Will Dabney, and Rémi Munos. 2017. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*. PMLR, 449–458.

[3] Glen Berseth, Daniel Geng, Coline Devin, Dinesh Jayaraman, Chelsea Finn, and Sergey Levine. 2019. SMiRL: Surprise Minimizing RL in Entropic Environments. (2019).

[4] Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. 2012. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics* 9, 1 (2012), 427–438.

[5] Jerry Zikun Chen. 2020. Reinforcement Learning Generalization with Surprise Minimization. *arXiv preprint arXiv:2004.12399* (2020).

[6] Christian Schroeder de Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. 2020. Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge? *arXiv preprint arXiv:2011.09533* (2020).

[7] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. 2019. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International Conference on Machine Learning*. PMLR, 3040–3049.

[8] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. 2019. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374* (2019).

[9] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[10] Jakub Grudzien Kuba, Ruiqing Chen, Munning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. 2021. Trust region policy optimisation in multi-agent reinforcement learning. *arXiv preprint arXiv:2109.11251* (2021).

[11] Iou-Jen Liu, Unnat Jain, Raymond A Yeh, and Alexander Schwing. 2021. Cooperative exploration for multi-agent deep reinforcement learning. In *International Conference on Machine Learning*. PMLR, 6826–6836.

[12] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *arXiv preprint arXiv:1706.02275* (2017).

[13] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. 2019. Maven: Multi-agent variational exploration. *arXiv preprint arXiv:1910.07483* (2019).

[14] Carlos Martin and Tuomas Sandholm. 2020. Efficient exploration of zero-sum stochastic games. *arXiv preprint arXiv:2002.10524* (2020).

[15] Bei Peng, Tabish Rashid, Christian A Schroeder de Witt, Pierre-Alexandre Kamienny, Philip HS Torr, Wendelin Böhmer, and Shimon Whiteson. 2020. FACMAC: Factored Multi-Agent Centralised Policy Gradients. *arXiv preprint arXiv:2003.06709* (2020).

[16] Adolfo Perrusquía, Wen Yu, and Xiaoou Li. 2021. Multi-agent reinforcement learning for redundant robot control in task-space. *International Journal of Machine Learning and Cybernetics* 12, 1 (2021), 231–241.

[17] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*. PMLR, 4295–4304.

[18] DJ Strouse, Max Kleiman-Weiner, Josh Tenenbaum, Matt Botvinick, and David J Schwab. 2018. Learning to Share and Hide Intentions using Information Regularization. *Advances in Neural Information Processing Systems* 31 (2018), 10249–10259.

[19] Karush Suri. 2021. Surprise Minimizing Multi-Agent Learning with Energy-based Models. (2021).

[20] Tonghan Wang, Jianhao Wang, Yi Wu, and Chongjie Zhang. 2019. Influence-based multi-agent exploration. *arXiv preprint arXiv:1910.05512* (2019).

[21] Zhao Xu, Yang Lyu, Quan Pan, Jinwen Hu, Chunhui Zhao, and Shuai Liu. 2018. Multi-vehicle flocking control with deep deterministic policy gradient method. In *2018 IEEE 14th International Conference on Control and Automation (ICCA)*. IEEE, 306–311.

[22] Tianpei Yang, Hongyao Tang, Chenjia Bai, Jinyi Liu, Jianye Hao, Zhaopeng Meng, and Peng Liu. 2021. Exploration in deep reinforcement learning: A comprehensive survey. *arXiv preprint arXiv:2109.06668* (2021).

[23] C. Yu, A. Velu, E. Vinitsky, Y. Wang, and Y. Wu. 2021. The Surprising Effectiveness of MAPPO in Cooperative, Multi-Agent Games. (2021).

[24] Zheqing Zhu, Erdem Bıyık, and Dorsa Sadigh. 2020. Multi-agent safe planning with gaussian processes. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 6260–6267.