A Variational Approach to Mutual Information-Based Coordination for Multi-Agent Reinforcement Learning

Woojun Kim
KAIST
Daejeon, Korea
woojun.kim@kaist.ac.kr

Whiyoung Jung
KAIST
Daejeon, Korea
wy.jung@kaist.ac.kr

Myungsik Cho
KAIST
Daejeon, Korea
ms.cho@kaist.ac.kr

Youngchul Sung
KAIST
Daejeon, Korea
ycsung@kaist.ac.kr

ABSTRACT

In this paper, we propose a new mutual information (MMI) framework for multi-agent reinforcement learning (MARL) to enable multiple agents to learn coordinated behaviors by regularizing the accumulated return with the simultaneous mutual information between multi-agent actions. By introducing a latent variable to induce nonzero mutual information between multi-agent actions and applying a variational bound, we derive a tractable lower bound on the considered MMI-regularized objective function. The derived tractable objective can be interpreted as maximum entropy reinforcement learning combined with uncertainty reduction of other agents' actions. Applying policy iteration to maximize the derived lower bound, we propose a practical algorithm named variational maximum mutual information multi-agent actor-critic (VM3-AC), which follows centralized learning with decentralized execution (CTDE). We evaluated VM3-AC for several games requiring coordination, and numerical results show that VM3-AC outperforms other MARL algorithms in multi-agent tasks requiring high-quality coordination.

KEYWORDS

Multi-Agent Reinforcement Learning; Coordination; Mutual Information

ACM Reference Format:
Woojun Kim, Whiyoung Jung, Myungsik Cho, and Youngchul Sung. 2023. A Variational Approach to Mutual Information-Based Coordination for Multi-Agent Reinforcement Learning. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

With the success of RL in the single-agent domain [13, 18], MARL is being actively studied and applied to real-world problems such as traffic control systems and connected self-driving cars, which can be modeled as multi-agent systems requiring coordinated control [1, 12]. The simplest approach to MARL is independent learning, which trains each agent independently while treating other agents as a part of the environment, but this approach suffers from the problem of non-stationarity of the environment. A common solution to this problem is to use fully-centralized critic in the framework of centralized training with decentralized execution (CTDE) [8, 11, 16, 20, 21]. For example, MADDPG [16] uses a centralized critic to train a decentralized policy for each agent, and COMA [4] uses a common centralized critic to train all decentralized policies. However, these approaches assume that decentralized policies are independent and hence the joint policy is the product of each agent's policy. Such non-correlated factorization of the joint policy limits the agents to learn coordinated behavior due to negligence of the influence of other agents [3, 26]. Recently, *mutual information* (MI) between multiple agents' actions has been considered as an effective intrinsic reward to promote coordination in MARL [10]. In [10], MI between agents' actions is captured as social influence and the goal is to maximize the sum of accumulated return and social influence between agents' actions. It is shown that the social influence approach is effective for sequential social dilemma games. In this framework, however, causality between actions under coordination is required, and it is not straightforward to coordinate multi-agents' simultaneous actions. In certain multi-agent games, coordination of simultaneous actions of multiple agents is required to achieve cooperation for a common goal. For example, suppose that a pack of wolves tries to catch a prey. To catch the prey, coordinating simultaneous actions among the wolves is more effective than coordinating one wolf's action and other wolves' actions at the next time because the latter case causes delay in coordination. In this paper, we propose a new approach to the MI-based coordination for MARL to coordinate simultaneous actions among multiple agents under the assumption of the knowledge of timing information among agents. Our approach is based on introducing a common latent variable to induce MI among simultaneous actions of multiple agents and on a variational lower bound on MI that enables tractable optimization. Under the proposed formulation, applying policy iteration by redefining value functions, we propose the VM3-AC algorithm for MARL to learn coordination of simultaneous actions among multiple agents. Numerical results show its superior performance on cooperative multi-agent tasks requiring coordination.

2 RELATED WORK

MI is a measure of dependence between two variables [2] and has been considered as an effective intrinsic reward for MARL [10, 24].

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.

[19] proposed an intrinsic reward for empowerment by maximizing MI between agent's action and its future state. [24] proposed two intrinsic rewards capturing the influence based on a decision-theoretic measure and MI between an agent's current actions/states and other agents' next states. In particular, [10] proposed a social influence intrinsic reward, which basically captures the mutual information between multiple agents' actions to achieve coordination, and showed that the social influence formulation yields good performance in sequential social dilemma environments. The difference of our approach from the social influence to MI-based coordination will be explained in Section 3 and Section 4.1.

Some previous works approached correlated policies from different perspectives. [15] proposed explicit modeling of correlated policies for multi-agent imitation learning, and [26] proposed a recursive reasoning framework for MARL to maximize the expected return by decomposing the joint policy into own policy and opponents' policies. Going beyond adopting correlated policies, our approach maximizes the MI between multiple agents' actions which is a measure of correlation.

In our approach, the MI between agents' action distributions is decomposed as the sum of each agent's action entropy and a variational term related to prediction of other agents' actions. Hence, our framework can be interpreted as enhancing correlated exploration by increasing the entropy of own policy [7] while decreasing the uncertainty about other agents' actions. Some previous works proposed other techniques to enhance correlated exploration [17, 28]. MAVEN addressed the poor exploration problem of QMIX by maximizing the mutual information between the latent variable and the observed trajectories [17]. However, MAVEN does not consider the correlation among policies.

3 BACKGROUND

Setup We consider a Markov Game [14], which is an extention of Markov Decision Process (MDP) to multi-agent setting. An N-agent Markov game is defined by an environment state space $\mathcal{S}$, action spaces for N agents $\mathcal{A}_1, \cdots, \mathcal{A}_N$, a state transition probability $p_{\mathcal{T}} : \mathcal{S} \times \boldsymbol{\mathcal{A}} \times \mathcal{S} \to [0, 1]$, where $\boldsymbol{\mathcal{A}} = \prod_{i=1}^{N} \mathcal{A}_i$ is the joint action space, and a reward function $\mathcal{R} : \mathcal{S} \times \boldsymbol{\mathcal{A}} \to \mathbb{R}$. At each time step t, Agent i with policy π^i executes action $a_t^i \in \mathcal{A}_i$ based on state $s_t \in \mathcal{S}$. The actions of all agents $\boldsymbol{a}_t = (a_t^1, \cdots, a_t^N)$ yield the next state s_{t+1} according to $p_{\mathcal{T}}$ and shared common reward r_t according to $\mathcal{R}$ under the assumption of fully-cooperative MARL. The discounted return is defined as $R_t = \sum_{n=t}^{\infty} \gamma^n r_n$, where $\gamma \in [0, 1)$ is the discounting factor.

We assume CTDE incorporating the resource asymmetry between training and execution phases, widely considered in MARL [4, 8, 16]. Under CTDE, each agent can access all information including the environment state, observations and actions of other agents in the training phase, whereas the policy of each agent is conditioned only on its own observation o_t^i in the execution phase. The goal of fully cooperative MARL is to find the optimal joint policy $\boldsymbol{\pi}^*$ that maximizes the objective $J(\boldsymbol{\pi}) = E_{\tau_0 \sim \boldsymbol{\pi}}[R_0]$, where $\tau_t = (s_t, \boldsymbol{a}_t, s_{t+1}, \boldsymbol{a}_{t+1}, \cdots)$ and $\boldsymbol{\pi} = (\pi^1, \cdots, \pi^N)$ denotes the joint policy of all agents.

Mutual Information-Based Coordination for MARL MI between agents' actions has been considered as an intrinsic reward to promote coordination in MARL [10]. Under this framework, one basically aims to find the policy that maximizes the weighted sum of the return and the MI between multi-agent actions. Thus, the MI-regularized objective function for joint policy $\boldsymbol{\pi}$ is given by

$$J(\boldsymbol{\pi}) = \mathbb{E}_{\tau_0 \sim \boldsymbol{\pi}}\left[\sum_{t=0}^{\infty} \gamma^t \Big(r_t + \alpha \sum_{(i,j)|i \neq j} I(a_t^i; a_t^j | s_t) \Big)\right], \tag{1}$$

where $I(a_t^i; a_t^j | s_t)$ is the MI between $a_t^i \sim \pi^i(\cdot|s_t)$ and $a_t^j \sim \pi^j(\cdot|s_t)$, and α is the temperature parameter that controls the relative importance of the MI against the reward. It is known that by regularization with MI in the objective function (1), the policy of each agent is encouraged to coordinate with other agents' policies. There are several approaches to implement (1). Under the social influence framework in [10], the MI is decomposed as

$$I(a_t^i; a_t^j | s_t) = \int_{a_t^i, a_t^j} p(a_t^i, a_t^j | s_t) \log \frac{p(a_t^i, a_t^j | s_t)}{p(a_t^i | s_t) p(a_t^j | s_t)} \tag{2}$$

$$= \int_{a_t^i} p(a_t^i | s_t) \int_{a_t^j} p(a_t^j | a_t^i, s_t) \log \frac{p(a_t^j | a_t^i, s_t)}{p(a_t^j | s_t)} \tag{3}$$

$$= \int_{a_t^i} p(a_t^i | s_t) \underbrace{D_{KL}(p(a_t^j | a_t^i, s_t) || p(a_t^j | s_t))}_{\triangleq \text{ social influence of agent } i \text{ on agent } j}, \tag{4}$$

where $D_{KL}(\cdot||\cdot)$ is the Kullback-Leibler divergence. In this decomposition, influencing Agent i's policy is given by $\pi^i = p(a_t^i | s_t)$ and influenced Agent j's policy is given by $\pi^j = p(a_t^j | a_t^i, s_t)$. The social influence is defined as the difference between $p(a_t^j | a_t^i, s_t)$ and $p(a_t^j | s_t)$. Hence, at time step t, influencing Agent i acts first and then influenced Agent j acts based on a_t^i after Agent i acts, as shown in Fig. 1(a). This sequential dependence between actions prevents multiple agents from performing simultaneous actions, which is an assumption of most decentralized execution. In addition, the social influence approach needs a strategy for action ordering because it divides all agents into a set of influencers and a set of influencees. One way to remove this action ordering is to model other agents [10]. In this case, the causal influence of action a_t^i of Agent i at time t on action a_{t+1}^j of Agent j at time $t+1$ is considered, as shown in Fig. 1(b), i.e., the social influence $D_{KL}(p(a_{t+1}^j | a_t^i, s_t) || p(a_{t+1}^j | s_t))$ instead of the influence term in (4)

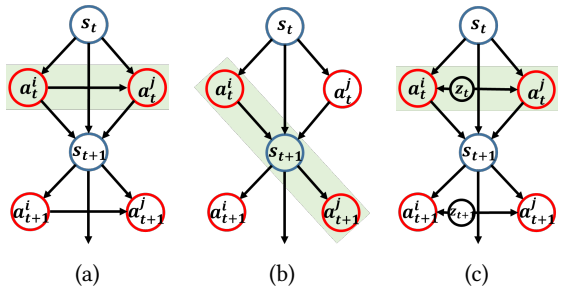


Figure 1: Causal diagram: (a) basic social influence, (b) social influence of modeling other agents, and (c) the proposed approach

is considered based on modeling $p(a^j_{t+1}|a^i_t, s_t)$ so that actions a^i_t and a^j_t can be performed simultaneously without ordering. In this case, however, the actually considered MI is $I(a^i_t; a^j_{t+1}|s_t)$ and is not the MI between a^i_t and a^j_t occurring at the same time. In this paper, we propose a different approach to MI regularization which enables simultaneous coordination between actions a^i_t and a^j_t both at time t without action ordering.

4 THE PROPOSED APPROACH

We assume that the environment is fully observable, i.e., each agent can observe the environment state s_t for theoretical development in this section, and will consider partially observable environment for practical algorithm construction under CTDE in the next section.

4.1 Formulation

Without explicit dependency between actions, $\pi^i(a^i_t|s_t)$ and $\pi^j(a^j_t|s_t)$ are conditionally independent for given environment state s_t and consequently the mutual information is always zero, i.e., $I(\pi^i(\cdot|s_t); \pi^j(\cdot|s_t)) = 0$. Then, the MI-regularized objective function (1) reduces to the standard MARL objective of only the accumulated return. In order to circumvent this difficulty, we propose a novel method to induce MI between actions. Our approach for inducing MI between concurrent two actions a^i_t and a^t_j of Agents i and j at time t is to introduce a latent variable Z_t, as shown in Fig. 1(c). We assume that the latent variable Z_t has a prior distribution $p_Z(z_t)$ and that actions a^i_t and a^j_t are generated from the state variable s_t and the latent random variable Z_t. Thus, Agent i's action a^i_t at time t is drawn from the policy distribution of Agent i as

$$a^i_t \sim \pi^i(\cdot|S_t = s_t, Z_t),\ i = 1, 2, \cdots, N, \tag{5}$$

where we use the upper case for random variables and the lower case for their realizations in the conditioning input terms for notational clarification. Then, even in case of deterministic policy, there is randomness in a^i_t for given $S_t = s_t$ due to the random input Z_t since a function of random variable is a random variable. In case of stochastic policy, there is additional randomness in a^i_t for given $S_t = s_t$ due to stochasticity of the policy itself. One can view the randomness due to Z_t as a perturbation to nominal a^i_t for given $S_t = s_t$. With the common perturbation-inducing variable Z_t to all agents' policies, two random variables a^t_i and a^t_j conditioned on $S_t = s_t$ are correlated due to common Z_t, and then nonzero MI $I(a^i_t; a^j_t|s_t)$ between concurrent a^i_t and a^j_t is induced. We aim to exploit this correlation for action coordination and correlated exploration in the training phase. (See Appendix A for a simple example and explanation of our basic idea with the simple example.)

With nontrivial MI $I(a^i_t; a^j_t|s_t)$, we now express this MI. First, note in (4) that we need $p(a^j_t|a^i_t, s_t)$ to compute the MI but we do not want to use $p(a^j_t|a^i_t, s_t)$ directly because $p(a^j_t|a^i_t, s_t)$ requires Agent j to know the action a^i_t of Agent i. For this, we adopt a variational distribution $q(a^j_t|a^i_t, s_t)$ to estimate $p(a^j_t|a^i_t, s_t)$ and derive a lower bound on the MI $I(a^i_t; a^j_t|s_t)$ as follows:

$$\begin{aligned}
I(a^i_t; a^j_t|s_t) &= \int_{a^i_t, a^j_t} p(a^i_t, a^j_t|s_t) \log \frac{p(a^i_t, a^j_t|s_t)}{p(a^i_t|s_t)p(a^j_t|s_t)} \\
&= \int_{a^i_t, a^j_t} p(a^i_t, a^j_t|s_t) \log \frac{p(a^i_t|s_t)p(a^j_t|a^i_t, s_t)q(a^j_t|a^i_t, s_t)}{p(a^i_t|s_t)p(a^j_t|s_t)q(a^j_t|a^i_t, s_t)} \\
&= \mathbb{E}_{p(a^i_t, a^j_t|s_t)} \left[\log \frac{q(a^j_t|a^i_t, s_t)}{p(a^j_t|s_t)} \right] \\
&\quad \times \mathbb{E}_{p(a^i_t|s_t)} \left[D_{KL}(p(a^j_t|a^i_t, s_t) \| q(a^j_t|a^i, s_t)) \right] \\
&\geq H(a^j_t|s_t) + \mathbb{E}_{p(a^i_t|s_t)p(a^j_t|a^i_t, s_t)} \left[\log q(a^j_t|a^i_t, s_t) \right],
\end{aligned} \tag{6}$$

where the last inequality in (6) holds because the KL divergence is always non-negative. Note that $H(a^j_t|s_t)$ is the entropy of a^j_t given s_t, i.e., the entropy of the following marginal distribution of a^j_t in our case:

$$\tilde{\pi}^j(a^j_t|s_t) := \int_{z_t} \pi^j(a^j_t|S_t = s_t, Z = z_t) p_Z(z_t) dz_t. \tag{7}$$

For the variational distribution $q(a^j_t|a^i_t, s_t)$ we consider a class of distributions $\boldsymbol{Q}$, i.e., $q(a^j_t|a^i_t, s_t) \in \boldsymbol{Q}$. The lower bound (6) becomes tight when $q(a^j_t|a^i_t, s_t)$ approximates $p(a^j_t|a^i_t, s_t)$ well, i.e., $D_{KL}(p(a^j_t|a^i_t, s_t) \| q(a^j_t|a^i, s_t))$ is small. Note that in our expansion, the lower bound on the MI $I(a^i_t; a^j_t|s_t)$ is expressed as *the sum of the action entropy* $H(a^j_t|s_t)$ *and the negative of the cross entropy of* $q(a^j_t|a^i_t, s_t)$ *relative to* $p(a^j_t|a^i_t, s_t)$ averaged over $p(a^i_t|s_t)$. Using the symmetry of MI, we can rewrite the lower bound as

$$\begin{aligned}
I(a^i_t; a^j_t|s_t) \geq \frac{1}{2} \Big\{ & H(a^i_t|s_t) + H(a^j_t|s_t) \\
& + \mathbb{E}_{p(a^i_t, a^j_t|s_t)} \left[\log q(a^j_t|a^i_t, s_t) + \log q(a^i_t|a^j_t, s_t) \right] \Big\}.
\end{aligned} \tag{8}$$

Then, our goal is to maximize this lower bound of MI by using a tractable approximation $q(a^i_t|a^j_t, s_t) \in \boldsymbol{Q}$. Our decompsition of MI based on the action entropy and the cross entropy is effective in our variational formulation for MI-based MARL. Consider one of the cross entropy terms in the right-hand side (RHS) of (8): $\mathbb{E}_{p(a^i_t, a^j_t|s_t)}[\log q(a^j_t|a^i_t, s_t)]$, which can be rewritten as

$$\begin{aligned}
\mathbb{E}_{p(a^i_t, a^j_t|s_t)}[\log q(a^j_t|a^i_t, s_t)] = -\mathbb{E}_{p(a^i_t|s_t)} \Big[& H(p(a^j_t|a^i_t, s_t)) \\
& + D_{KL}(p(a^j_t|a^i_t, s_t) || q(a^j_t|a^i_t, s_t)) \Big],
\end{aligned} \tag{9}$$

based on the well-known decomposition of the cross entropy. Hence, by maximizing this cross entropy term, due to the negation in (9) we can learn π^i (generating a^i_t) and π^j (generating a^j_t) so that the conditional entropy $H(p(a^j_t|a^i_t, s_t))$ of a^j_t given a^i_t is minimized, i.e., the two actions are more correlated to each other, and learn q that closely approximates the true $p(a^j_t|a^i_t, s_t)$, i.e., the D_{KL} term in (9) is minimized.

4.2 Modified Policy Iteration

Our algorithm construction is based on policy iteration. In order to develop policy iteration for the proposed MI framework, we

first replace the original MI-regularized objective function (1) with the following tractable objective function based on the variational lower bound (8):

$$\hat{J}(\boldsymbol{\pi}, q) = \mathbb{E}_{\substack{\tau_0 \sim \boldsymbol{\pi} \\ z_t \sim p_Z}} \Bigg[\sum_{t=0}^{\infty} \gamma^t \Big(r_t(s_t, \boldsymbol{a_t}) + \alpha N \sum_{i=1}^{N} H(a_t^i | s_t) + \alpha \sum_{i=1}^{N} \sum_{j \neq i} \log q(a_t^j | a_t^i, s_t) \Big) \Bigg], \quad (10)$$

where $\boldsymbol{\pi} = [\pi^1, \cdots, \pi^N]$ and π^i is given by (5) and $\boldsymbol{a}_t = [a_t^1, \cdots, a_t^N]$. Then, we determine the individual objective function $\hat{J}^i(\pi^i, q)$ for Agent i as the sum of the terms in (10) associated with Agent i's policy π^i or action a_t^i, given by

$$\hat{J}^i(\pi^i, q) = \mathbb{E}_{\substack{\tau_0 \sim \boldsymbol{\pi} \\ z_t \sim p_Z}} \Bigg[\sum_{t=0}^{\infty} \gamma^t \Big(\underbrace{r_t(s_t, \boldsymbol{a_t}) + \beta \cdot H(a^i | s_t)}_{(a)} + \frac{\beta}{N} \sum_{j \neq i} \underbrace{\Big[\log q(a_t^i | a_t^j, s_t) + \log q(a_t^j | a_t^i, s_t) \Big]}_{(b)} \Big) \Bigg], \quad (11)$$

where $\beta = \alpha N$ is the temperature parameter. Note that maximizing the term (a) in (11) implies that each agent maximizes the weighted sum of the return and the action entropy, which can be interpreted as an extension of maximum entropy RL [7] to multi-agent setting. On the other hand, maximizing the term (b) with respect to π^i and q means that we update the policy π^i so that the conditional entropy of a_t^j given a_t^i and the conditional entropy of a_t^i given a_t^j are reduced, as already mentioned below (9). Thus, the objective function (11) can be interpreted as the *maximum entropy MARL objective combined with action correlation or coordination*. Hence, the proposed objective function (11) can be considered as one implementation of the concept of *correlated exploration* in MARL [17].

Now, in order to learn policy π^i to maximize the objective function (11), we modify the policy iteration in standard RL. For this, we redefine the value functions for Agent i as

$$Q_i^{\boldsymbol{\pi}}(s, a) \triangleq \mathbb{E}_{\substack{\tau_0 \sim \boldsymbol{\pi} \\ z_t \sim p_Z}} \Big[r_0 + \gamma V_i^{\boldsymbol{\pi}}(s_1) \Big| s_0 = s, \boldsymbol{a}_0 = \boldsymbol{a} \Big], \quad (12)$$

$$V_i^{\boldsymbol{\pi}}(s) \triangleq \mathbb{E}_{\substack{\tau_0 \sim \boldsymbol{\pi} \\ z_t \sim p_Z}} \Bigg[\sum_{t=0}^{\infty} \gamma^t \Big(r_t + \beta H(a_t^i | s_t) + \frac{\beta}{N} \sum_{j \neq i} \log q^{(i,j)}(a_t^i, a_t^j | s_t) \Big) \Big| s_0 = s \Bigg], \quad (13)$$

where $q^{(i,j)}(a_t^i, a_t^j | s_t) \triangleq q(a_t^i | a_t^j, s_t) q(a_t^j | a_t^i, s_t)$. Then, the Bellman operator corresponding to $V_i^{\boldsymbol{\pi}}$ and $Q_i^{\boldsymbol{\pi}}$ on the value function estimates $V_i(s)$ and $Q_i(s, \boldsymbol{a})$ is given by

$$\mathcal{T}^{\boldsymbol{\pi}} Q_i(s, \boldsymbol{a}) \triangleq r(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s' \sim p}[V_i(s')], \text{ where} \quad (14)$$

$$V_i(s) = \mathbb{E}\Big[Q_i(s, \boldsymbol{a}) - \beta \log \tilde{\pi}^i(a^i | s) + \tfrac{\beta}{N} \textstyle\sum_{j \neq i} \log q^{(i,j)}(a^i, a^j | s) \Big], \text{ and}$$

$\tilde{\pi}^i$ is the marginal distribution given in (7). In the policy evaluation step, we compute the value functions (12) and (13) by applying the modified Bellman operator $\mathcal{T}^{\boldsymbol{\pi}}$ repeatedly to an initial function $Q_i^{(0)}$.

Proposition 1. *(Variational Policy Evaluation). For fixed $\boldsymbol{\pi}$ and the variational distribution q, consider the modified Bellman operator $\mathcal{T}^{\boldsymbol{\pi}}$ in (14) and an arbitrary initial function $Q_i^{(0)} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and define $Q_i^{(k+1)} = \mathcal{T}^{\boldsymbol{\pi}} Q_i^{(k)}$. Then, $Q_i^{(k)}$ converges to $Q_i^{\boldsymbol{\pi}}$ defined in (12).*

Proof. See Appendix B.

In the policy improvement step, we update the policy and the variational distribution by using the value function evaluated in the policy evaluation step. Here, each agent updates its policy and variational distribution while keeping other agents' policies fixed as follows: $(\pi_{k+1}^i, q_{k+1}) =$

$$\underset{\pi^i, q}{\arg\max}\, \mathbb{E}_{\substack{(a^i, a^{-i}) \sim (\pi^i, \pi_k^{-i}) \\ z_k \sim P_Z}} \Bigg[Q_i^{\boldsymbol{\pi}_k}(s, \boldsymbol{a}) - \beta \log \tilde{\pi}^i(a^i | s) + \frac{\beta}{N} \sum_{j \neq i} \log q^{(i,j)}(a^i, a^j | s)) \Bigg], \quad (15)$$

where $a^{-i} \triangleq \{a^1, \cdots, a^N\} \backslash \{a^i\}$ and π_k^{-i} is the collection the policies for all agents except Agent i at the k-th iteration. Then, we have the following proposition regarding the improvement step.

Proposition 2. *(Variational Policy Improvement). Let π_{new}^i and q_{new} be the updated policy and the variational distribution from (15). Then, $Q_i^{\pi_{new}^i, \pi_{old}^{-i}}(s, \boldsymbol{a}) \geq Q_i^{\pi_{old}^i, \pi_{old}^{-i}}(s, \boldsymbol{a})$ for all $(s, \boldsymbol{a}) \in (\mathcal{S} \times \mathcal{A})$. Here, $Q_i^{\pi_{new}^i, \pi_{old}^{-i}}(s, \boldsymbol{a})$ means $Q_i^{\boldsymbol{\pi}}(s, \boldsymbol{a})|_{\boldsymbol{\pi} = (\pi_{new}^i, \pi_{old}^{-i})}$.*

Proof. See Appendix B.

The modified policy iteration is defined as applying the variational policy evaluation and variational improvement steps in an alternating manner. Each agent trains its policy, critic and the variational distribution to maximize its objective function (11).

5 ALGORITHM CONSTRUCTION

Summarizing the development above, we now propose the variational maximum mutual information multi-agent actor-critic (VM3-AC) algorithm, which can be applied to continuous and partially observable multi-agent environments under CTDE. The overall operation of VM3-AC is shown in Fig. 2. Under CTDE, each agent's policy is conditioned only on local observation, and centralized critics are conditioned on either the environment state or the observations of all agents, depending on the situation [16]. Let $\boldsymbol{x}$ denote either the environment state s or the observations of all agents $(o_1, \cdots, o_N)$, whichever is used. In order to deal with the large continuous state-action spaces, we adopt deep neural networks to approximate the required functions. For Agent i, we parameterize the policy as $\pi_{\phi^i}^i(a | o^i, z)$ with parameter ϕ^i, the variational distribution as $q_{\xi^i}(a^j | a^i, (o^i, o^j))$ with parameter ξ^i, the state-value function as $V_{\psi_i}^i(\boldsymbol{x})$ with parameter ψ^i, and two action-value functions as $Q_{\theta^{i,1}}^i(\boldsymbol{x}, \boldsymbol{a})$ and $Q_{\theta^{i,2}}^i(\boldsymbol{x}, \boldsymbol{a})$ with parameters $\theta^{i,1}$ and $\theta^{i,2}$. Note that in the original variational distribution, a_t^j is conditioned on a_t^i and s_t. In the partially observable case, we replace s_t with (o_i, o_j).

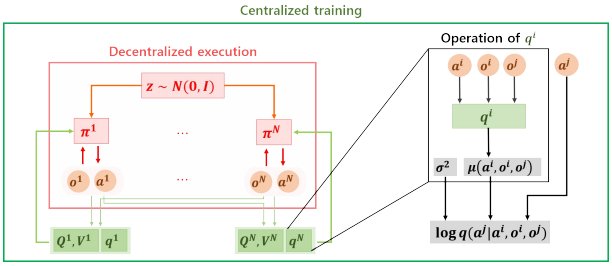


Figure 2: Overall operation of the proposed VM3-AC. We only need the operation in the red box after training.

For the prior distribution P_Z of the injection variable z_t, we use zero-mean multivariate Gaussian distribution with identity covariance matrix, i.e., $z_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, where the dimension is a hyperparameter, given in Appendix E. We further assume that the class $\mathcal{Q}$ of the variational distribution is multivariate Gaussian distribution with constant covariance matrix $\sigma^2\mathbf{I}$ with dimension of the action dimension, i.e., $\mathcal{Q} = \{q_{\xi^i}(a^j|a^i, (o^i, o^j)) = \mathcal{N}(\mu_{\xi^i}(a^i, o^i, o^j), \sigma^2\mathbf{I})\}$, where $\mu_{\xi^i}(a^i, o^i, o^j)$ is the mean of the distribution.

Centralized Training The parameterized value functions, the policy, and the variational distribution are trained based on proper loss functions derived from Section 4.2 in a similar way to the training in SAC in a centralized manner. During the centralized training, correlated exploration works so as to find a good set of joint policies of the N agents due to our common injection variable as explained in Section 4. Now, we provide the training details and pseudo code.

The value functions $V^i_{\psi_i}(\boldsymbol{x})$, $Q^i_{\theta_i}(\boldsymbol{x}, \boldsymbol{a})$ are updated based on the modified Bellman operator defined in (13) and (14). The state-value function $V^i_{\psi_i}(\boldsymbol{x})$ is trained to minimize the following loss function:

$$\mathcal{L}_V(\psi^i) = \mathbb{E}_{s_t \sim D}\left[\frac{1}{2}(V^i_{\psi^i}(\boldsymbol{x}_t) - \hat{V}^i_{\psi^i}(\boldsymbol{x}_t))^2\right] \tag{16}$$

where D is the replay buffer that stores the transitions $(\boldsymbol{x}_t, \boldsymbol{a}_t, r_t, \boldsymbol{x}_{t+1})$; $Q^i_{min}(\boldsymbol{x}_t, a^i_t) = \min[Q^i_{\theta^{i,1}}(\boldsymbol{x}_t, a^i_t), Q^i_{\theta^{i,2}}(\boldsymbol{x}_t, a^i_t)]$ is the minimum of the two action-value functions to prevent the overestimation problem [5]; and

$$\hat{V}^i_{\psi^i}(\boldsymbol{x}_t) = \mathbb{E}_{z_t \sim N(0,\boldsymbol{I}), \{a^k \sim \pi^k(\cdot|o^k_t, z_t)\}_{k=1}^N}\Bigg[Q^i_{min}(\boldsymbol{x}_t, \boldsymbol{a}_t) - \beta \log \pi^i_{\phi^i}(a^i_t|o^i_t, z_t) + \frac{\beta}{N}\sum_{j\neq i} \log q^{(i,j)}_{\xi^i}(a^i_t, a^j_t|o^i_t, o^j_t)\Bigg]. \tag{17}$$

Note that in the second term of the RHS of (17), originally we should have used the marginalized version, $-\beta \log \tilde{\pi}^i_{\phi^i}(a^i_t|o^i_t) = -\beta \log \mathbb{E}_{z_t \sim N(0,\boldsymbol{I})}[\pi^i_{\phi^i}(a^i_t|o^i_t, z_t)]$. However, for simplicity of computation, we took the expectation $\mathbb{E}_{z_t \sim N(0,\boldsymbol{I})}$ outside the logarithm. Hence, there exists Jensen's inequality type approximation error. We observe that this approximation works well.

The two action-value functions are updated by minimizing the loss

$$\mathcal{L}_Q(\theta^i) = \mathbb{E}_{(\boldsymbol{x}_t, \boldsymbol{a}_t) \sim D}\left[\frac{1}{2}(Q_{\theta^i}(\boldsymbol{x}_t, \boldsymbol{a}_t) - \hat{Q}(\boldsymbol{x}_t, \boldsymbol{a}_t))^2\right] \tag{18}$$

where

$$\hat{Q}(\boldsymbol{x}_t, \boldsymbol{a}_t) = r_t(x_t, \boldsymbol{a}_t) + \gamma \mathbb{E}_{\boldsymbol{x}_{t+1}}[V^i_{\bar{\psi}}(x_{t+1})] \tag{19}$$

and $V^i_{\bar{\psi}}$ is the target value network, which is updated by the exponential moving average method. We implement the reparameterization trick to estimate the stochastic gradient of policy loss. Then, the action of agent i is given by $a^i = f_{\phi^i}(s; \epsilon^i, z)$, where $\epsilon^i \sim \mathcal{N}(0, \boldsymbol{I})$ and $z \sim \mathcal{N}(0, \boldsymbol{I})$. The policy for Agent i and the variational distribution are trained to minimize the following policy improvement loss, $\mathcal{L}_{\pi^i,q}(\phi^i, \xi)$

$$= \mathbb{E}_{\substack{s_t \sim D, \\ \epsilon^i \sim \mathcal{N}, \\ z \sim \mathcal{N}}}\Bigg[-Q^i_{\theta^{i,1}}(\boldsymbol{x}_t, \boldsymbol{a}) + \beta \log \pi^i_{\phi^i}(a^i|o^i_t, z) - \frac{\beta}{N}\sum_{j\neq i} \log q^{(i,j)}_{\xi^i}(\pi^i_{\phi^i}(a^i|o^i_t, z), \pi^j_{\phi^j}(a^j|o^j_t, z)|o^i_t, o^j_t)\Bigg] \tag{20}$$

where $q^{(i,j)}_{\xi^i}(\pi^i_{\phi^i}(a^i|o^i_t, z), \pi^j_{\phi^j}(a^j|o^j_t, z)|o^i_t, o^j_t)$

$$= \underbrace{q_{\xi^i}(\pi^i_{\phi^i}(a^i|o^i_t, z)|\pi^j_{\phi^j}(a^j|o^j_t, z)|o^i_t, o^j_t)}_{(a)} \times \underbrace{q_{\xi^i}(\pi^j_{\phi^j}(a^j|o^j_t, z)|\pi^i_{\phi^i}(a^i|o^i_t, z)|o^i_t, o^j_t)}_{(b)}. \tag{21}$$

Again, for simplicity of computation, we took the expectation $\mathbb{E}_{z_t \sim N(0,\boldsymbol{I})}$ outside the logarithm for the second term in the RHS in (20). Since approximation of the variational distribution is not accurate in the early stage of training and the learning via the term (a) in (21) is more susceptible to approximation error, we propagate the gradient only through the term (b) in (21) to make learning stable. Note that minimizing $-\log q_{\xi^i}(a^j|a^i, s_t)$ is equivalent to minimizing the mean-squared error between a^j and $\mu_{\xi^i}(a^i, o^i, o^j)$ due to our Gaussian assumption on the variational distribution.

Decentralized Execution In the centralized training phase, we pick actions $(a^1_t, \cdots, a^N_t)$ according to $\pi^1(a^1_t|s_t, z_t), \cdots, \pi^N(a^N_t|s_t, z_t)$ (or with s_t replaced with $(o^1_t, \cdots, o^N_t)$), where common z_t generated from zero-mean Gaussian distribution is shared under the centralized assumption. However, in the decentralized execution phase, sharing common z_t requires communication among the agents. To remove this communication necessity, we consider two methods. First, under the assumption of synchronization, we can make all agents have the same Gaussian random sequence generator and distribute the same seed and initiation timing to this random sequence generator only once in the beginning of the execution phase. In other words, we require all agents to have the same Gaussian random sequence generator and distribute the same seed and initiation timing to these random sequence generators before deployment for the execution phase. (Mahajan et al. [17] also considered that multiple agents share the realization of latent variables in the beginning of the episode.) Second, we exploit the property of

Algorithm 1 VM3-AC (L=1)

Centralized training phase
Initialize parameter $\phi^i, \theta^i, \psi^i, \overline{\psi}^i, \xi^i, \ \forall i \in \{1, \cdots, N\}$
for $episode = 1, 2, \cdots$ **do**
  Initialize state s_0 and each agent observes o_0^i
  for $t < T$ and $s_t \neq$ terminal **do**
    Generate $z_t \sim \mathcal{N}(0, I)$ and select action $a_t^i \sim \pi^i(\cdot|o_t^i, z_t)$, $\forall i$
    Execute $\boldsymbol{a_t}$ and each agent i receives r_t and o_{t+1}^i
    Store transitions in D
  end for
  for each gradient step **do**
    Sample a minibatch from D and generate $z_l \sim \mathcal{N}(0, I)$ for each transition.
    Update θ^i, ψ^i by minimizing the loss (18) and (19)
    Update ϕ^i, ξ^i by minimizing the loss (20)
  end for
  Update $\overline{\psi}^i$ using the moving average method
end for

Decentralized execution phase
Initialize state s_0 and each agent observes o_0^i
for each environment step **do**
  Select action $a_t^i \sim \pi^i(\cdot|o_t^i, z_t)$ where $z_t = \overrightarrow{0}$ (or sample from the Gaussian random sequence generator with the same seed)
  Execute $\boldsymbol{a_t}$ and each agent i receives o_{t+1}^i
end for

zero-mean Gaussian input variable z_t to the policy network. During the centralized training period, the parameters $\phi^1, \cdots, \phi^N$ of the policy networks $\pi^1_{\phi^1}(a|o^1, z), \cdots, \pi^N_{\phi^N}(a|o^N, z)$ (with input (o^i, z) and output a) are learned so that actions $a_t^1, \cdots, a_t^N$ are coordinated for random perturbation input z_t drawn from P_Z. Note that the coordination behavior is learned and engraved into the parameters $\phi_1, \cdots, \phi_N$ not into the input z_t. So, we only use this stored parameter information during the decentralized execution phase. We apply the common mean value $\mathbb{E}\{z_t\}$ to the z_t input of the trained policy network $\pi^i_{\phi^i}(a_t^i|o_t^i, z_t)$ of Agent i, $\forall i$. In this case, actions $a_t^1, \cdots, a_t^N$ are independent conditioned on $s_t \ni (o_t^1, \cdots, o_t^N)$ but a specific joint bias (most representative joint bias) is applied to actions $a_t^1, \cdots, a_t^N$. We expect that this joint bias is helpful and this situation is described in a toy example in Appendix A. In this way, the proposed algorithm is fully operative under CTDE. The ablation study is provided in Sec. 6.

6 EXPERIMENT

In this section, we provide numerical results on both continuous and discrete action tasks.

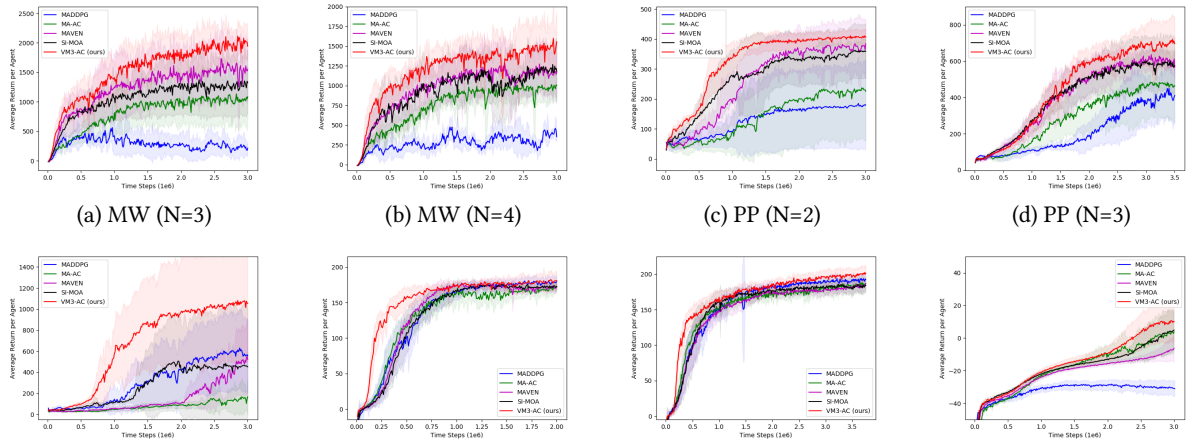
Experiment on continuous action tasks We consider the following continuous action tasks with the varying number of agents: multi-walker [6], predator-prey [16], cooperative treasure collection [9], and cooperative navigation [16]. The detailed setting of each task is provided in Appendix F. Here, we considered four baselines: 1) MADDPG [16] - an extension of DDPG with a centralized critic to train a decentralized policy for each agent. 2) Multi-agent actor-critic (MA-AC) - a variant of VM3-AC ($\beta = 0$) without the latent variable. 3) Multi-agent variational exploration (MAVEN) [17]. Similarly to VM3-AC, MAVEN introduced latent variable and variational approach for optimizing the mutual information. However, MAVEN does not consider the mutual information between actions but considers the mutual information between the latent variable and trajectories of the agents. 4) Social Influence with MOA (SI-MOA) [10], which is explained in Section 3. Both MAVEN and SI-MOA are implemented on top of MA-AC since we consider continuous action-space environments.

Fig. 3 shows the learning curves for the considered four environments with the different numbers of agents. The y-axis denotes the average of all agents' rewards averaged over 7 random seeds, and the x-axis denotes the time step. The hyperparameters including the temperature parameter β and the dimension of the latent variable are provided in Appendix E. As shown in Fig. 3, VM3-AC outperforms the baselines in the considered environments. Especially, in the case of the multi-walker environment, VM3-AC has a large performance gain over existing state-of-the-art algorithms. This is because the agents in the multi-walker environment are strongly required to learn simultaneous coordination in order to obtain high rewards. In addition, the agents in the predator-prey environment, where the number of agents is four, should spread out in groups of two to get more rewards. In this environment, VM3-AC also has a large performance gain. Thus, it is seen that the proposed MMI framework improves performance in complex multi-agent tasks requiring high-quality coordination. It is observed that both MAVEN and SI-MOA outperform the basic algorithm MA-AC but not VM3-AC. Hence, the numerical results show that the way of using MI by the proposed VM3-AC algorithm has some advantages over those by MAVEN and SI-MOA, especially for MARL tasks requiring coordination of concurrent actions.

Experiment on discrete action task We also considered the StarcraftII micromanagement benchmark (SMAC) environment [22]. We modified the SMAC environment to be sparse by giving rewards when an ally or an enemy dies and a time penalty. Thus, in the case of *3s vs 3z*, where we need to control three stalkers to beat the three zealots (enemy), the reward is hardly obtained because it takes a long time to remove a zealot. We provided the detailed setting of the modified SMAC environment in Appendix G. We considered five state-of-the-art baselines: DOP [25], FOP [27], LICA [29], MAVEN [17], and VDAC [23]. We implemented VM3-AC on the top of FOP by introducing the latent variable and replacing the entropy term in [27] with the MI. Fig. 4 shows the performances of VM3-AC and the baselines on three maps in SMAC. It is observed that VM3-AC outperforms the baselines. Especially on 3svs3z, in which reward is highly sparse, VM3-AC outperforms the baselines in terms of both training speed and final performance.

6.1 Ablation Study and Discussion

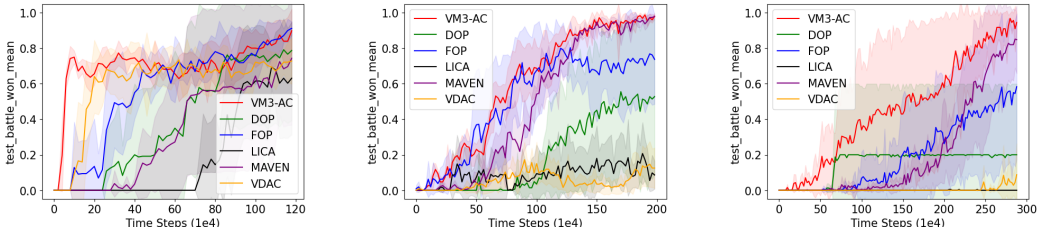
In this subsection, we provide ablation studies and discussion on the major techniques and hyperparameters of VM3-AC: 1) mutual information versus entropy 2) the latent variable, 3) the temperature parameter β, 4) injecting zero vector instead of the latent variable z to policies in the execution phase and 5) scalability.



(a) MW (N=3) (b) MW (N=4) (c) PP (N=2) (d) PP (N=3)

(e) PP (N=4) (f) CTC (N=4) (g) CTC (N=5) (h) CN (N=3)

Figure 3: Performance of MADDPG (blue), MA-AC (green), MAVEN (purple), SI-MOA (black), and VM3-AC (the proposed method, red) on multi-walker environments (a)-(b), predator-prey (c)-(e), cooperative treasure collection (f)-(g), and cooperative navigation (f). (MW, PP, CTC, and CN denote multi-walker, predator-prey, cooperative treasure collection and cooperative navigation, respectively)



(a) *3m* (b) *2s3z* (c) *3s vs 3z*

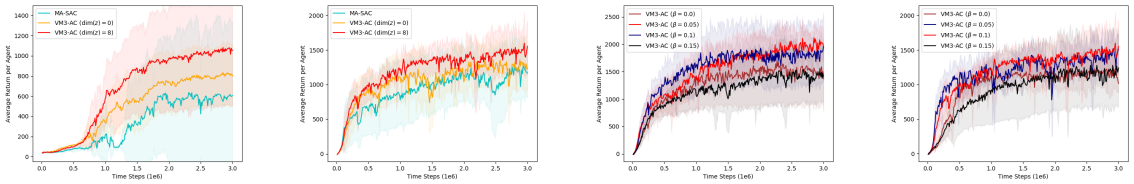
Figure 4: Performance of DOP (green), FOP (blue), LICA (black), MAVEN (purple), VDAC (orange) and VM3-AC (red) on three maps in the modified SMAC environment.

Mutual information versus entropy: The proposed MI framework maximizes the sum of the action entropy and the negative of the cross entropy of the variational conditional distribution relative to the true conditional distribution, which provides a lower bound of MI between actions. As aforementioned, maximizing the sum of the action entropy and the negative of the cross entropy of the variational conditional distribution relative to the true conditional distribution enhances exploration and predictability for other agents' actions. Hence, the proposed MI framework enhances correlated exploration among agents.

We compared VM3-AC with multi-agent-SAC (MA-SAC), which is an extension of maximum entropy soft actor-critic (SAC) [7] to multi-agent setting. For MA-SAC, we extended SAC to multi-agent settings in the manner of independent learning. Each agent trains its decentralized policy using decentralized critic to maximize the weighted sum of the cumulative return and the entropy of its policy. Adopting the framework of CTDE, we replaced decentralized critic with centralized critic which incorporates observations and actions of all agents.

We performed an experiment in the predator-prey environment with four agents where the number of required agents to catch the prey is two. In this environment, the agents started at the center of the map. Hence, the agents should spread out in the group of two to catch preys efficiently. Fig.6 shows the positions of the four agents at five time-steps after the episode starts. The first and second rows in Fig.6 show the results of VM3-AC and MA-SAC in the early stage of the training, respectively. It is seen that the agents of VM3-AC explore in the group of two while the agents of MA-SAC tend to explore independently. We provided the performance comparisons of VM3-AC with MA-SAC in Fig.5 (a) and (b).

Latent variable: The role of the latent variable is to induce MI among concurrent actions and inject an additional degree of freedom for action control. We compared VM3-AC and VM3-AC without the latent variable (implemented by setting dimension$(z_t) = 0$) in the multi-walker environment. In both cases, VM3-AC yields



(a) PP (N=4) (b) MW (N=4) (c) MW (N=3) (d) MW (N=4)

Figure 5: (a) and (b): VM3-AC (red), VM3-AC without latent variable (orange), and MA-SAC (cyan) and (c) and (d): performance with respect to the temperature parameter

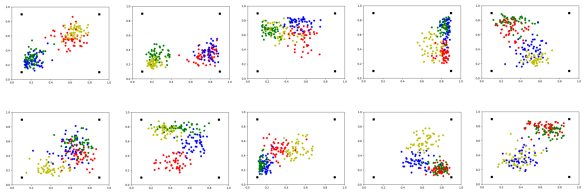


Figure 6: The positions of four agents after five time-steps after the episode begins in the early stage of the training: 1st row - VM3-AC and 2nd row - MA-SAC. The figures in column correspond to a different seed. The black squares are the preys and each color except black shows the position of each agent.

better performance than VM3-AC without the latent variable as shown in Fig.5(a) and 5(b). Here, the gain by VM3-AC without the latent variable (i.e., dimension$(z_t) = 0$) over MA-SAC is solely due to passive modeling $p(a_t^j|a_t^i, s_t)$ by using $q(a_t^j|a_t^i, s_t)$, not including active injection of coordination by z_t.

Injecting mean vector $\mathbb{E}\{z_t\}$ to the z_t-input of policy network $\pi^i_{\phi^i}(\cdot|o_t^i, z_t)$ during the execution phase: As mentioned in Section 5, we applied the mean vector of z_t, i.e., $\mathbb{E}\{z_t\}$ to the z_t-input of the policy deep neural network $\pi^i_{\phi^i}(\cdot|o_t^i, z_t)$ during the execution phase so as to execute actions without communication in the execution phase. We compared the performance of decentralized policies that use the mean vector $\mathbb{E}\{z_t\}$ and decentralized policies which use the latent variable z_t assuming communication. We used deterministic evaluation based on 20 episodes generated by the corresponding deterministic policy, i.e., each agent selects action using the mean network of Gaussian policy $\pi^i_{\phi^i}$. We averaged the return over 7 seeds, and the result is shown in Table 1. It is seen that the mean vector replacement method yields almost the same performance and enables fully decentralized execution without noticeable performance loss. Please see Appendix A for intuition.

Temperature parameter β: The role of temperature parameter β is to control the relative importance between the reward and the MI. We evaluated VM3-AC by varying $\beta = [0, 0.05, 0.1, 0.15]$ in the multi-walker environment with $N = 3$ and $N = 4$. Fig. 5(c) and 5(d) show that VM3-AC with the temperature value around $[0.05, 0.1]$ yields good performance.

Table 1: Impact of replacing the latent variable $z_t \sim \mathcal{N}(0, \mathbf{I})$ with mean vector $z_t = \mathbb{E}(z_t)$ in the execution phase

	PP (N=2)	PP (N=3)	PP (N=4)
$z_t \sim \mathcal{N}(0, \mathbf{I})$	413	734	1123
$z_t = \mathbb{E}(z_t)$	409	743	1147

Scalability: Many MARL algorithms which use a centralized critic such as MADDPG [16] can suffer from the problem of scalability due to increasing joint state-action space as the number of agents increases. VM3-AC can also suffer from the same issue but we can address the problem by adopting an attention mechanism as in MAAC [8]. Additionally, VM3-AC needs more variational approximation networks as the number of agents increases. As many MARL algorithms share the parameters among agents, we can share the parameters for the variational approximation networks. We expect that parameter sharing can handle the scalability of the proposed method.

7 CONCLUSION

In this paper, we have proposed a new approach to MI-based coordinated MARL to induce the coordination of concurrent actions under CTDE. In the proposed approach, a common correlation-inducing random variable is injected into each policy network, and the MI between actions induced by this variable is expressed as a tractable form by using a variational distribution. The derived objective consists of the maximum entropy RL combined predictability enhancement (or uncertainty reduction) for other agents' actions, which can be interpreted as correlated exploration. We evaluated the derived algorithm named VM3-AC on both continuous and discrete action tasks and the numerical results show that VM3-AC outperforms other state-of-the-art baselines, especially in multi-agent tasks requiring high-quality coordination among agents.

Limitation One can think sharing the common variable requires communication between agents. To handle this, we introduced two methods including sharing a Gaussian random sequence generator at the beginning of the episode and injecting the mean vector into the latent vector in the execution. Here, reference timing information on top of time step synchronization is required for the method of sharing a Gaussian random generator. This requirement of communication is one limitation of our work, but we provided an ablation study on this alternative and it was seen that the alternative performs well.

8 ACKNOWLEDGEMENTS

This work was partly supported by Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2022-0-00469) and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT). (NRF-2021R1A2C2009143)

REFERENCES

[1] CP Andriotis and KG Papakonstantinou. 2019. Managing engineering systems with large state and action spaces through deep reinforcement learning. *Reliability Engineering & System Safety* 191 (2019), 106483.
[2] T. M. Cover and J. A. Thomas. 2006. *Elements of Information Theory*. Wiley.
[3] Christian Schroeder de Witt, Jakob Foerster, Gregory Farquhar, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. 2019. Multi-Agent Common Knowledge Reinforcement Learning. In *Advances in Neural Information Processing Systems*. 9924–9935.
[4] Jakob N Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *Thirty-second AAAI conference on artificial intelligence*.
[5] Scott Fujimoto, Herke Van Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477* (2018).
[6] Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. 2017. Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*. Springer, 66–83.
[7] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290* (2018).
[8] Shariq Iqbal and Fei Sha. 2018. Actor-attention-critic for multi-agent reinforcement learning. *arXiv preprint arXiv:1810.02912* (2018).
[9] Shariq Iqbal and Fei Sha. 2019. Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning*. PMLR, 2961–2970.
[10] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro A Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. 2018. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. *arXiv preprint arXiv:1810.08647* (2018).
[11] Jeewon Jeon, Woojun Kim, Whiyoung Jung, and Youngchul Sung. 2022. Maser: Multi-agent reinforcement learning with subgoals generated from experience replay buffer. In *International Conference on Machine Learning*. PMLR, 10041–10052.
[12] Minne Li, Zhiwei Qin, Yan Jiao, Yaodong Yang, Jun Wang, Chenxi Wang, Guobin Wu, and Jieping Ye. 2019. Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. In *The World Wide Web Conference*. 983–994.
[13] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).
[14] Michael L Littman. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*. Elsevier, 157–163.
[15] Minghuan Liu, Ming Zhou, Weinan Zhang, Yuzheng Zhuang, Jun Wang, Wulong Liu, and Yong Yu. 2020. Multi-Agent Interactions Modeling with Correlated Policies. *arXiv preprint arXiv:2001.03415* (2020).
[16] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*. 6379–6390.
[17] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. 2019. MAVEN: Multi-Agent Variational Exploration. In *Advances in Neural Information Processing Systems*. 7611–7622.
[18] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.
[19] Shakir Mohamed and Danilo J Rezende. 2015. Variational information maximisation for intrinsically motivated reinforcement learning. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*. 2125–2133.
[20] Afshin OroojlooyJadid and Davood Hajinezhad. 2019. A review of cooperative multi-agent deep reinforcement learning. *arXiv preprint arXiv:1908.03963* (2019).
[21] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. *arXiv preprint arXiv:1803.11485* (2018).
[22] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. 2019. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043* (2019).
[23] Jianyu Su, Stephen Adams, and Peter A Beling. 2021. Value-decomposition multi-agent actor-critics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 11352–11360.
[24] Tonghan Wang, Jianhao Wang, Yi Wu, and Chongjie Zhang. 2019. Influence-based multi-agent exploration. *arXiv preprint arXiv:1910.05512* (2019).
[25] Yihan Wang, Beining Han, Tonghan Wang, Heng Dong, and Chongjie Zhang. 2020. Dop: Off-policy multi-agent decomposed policy gradients. In *International Conference on Learning Representations*.
[26] Ying Wen, Yaodong Yang, Rui Luo, Jun Wang, and Wei Pan. 2019. Probabilistic recursive reasoning for multi-agent reinforcement learning. *arXiv preprint arXiv:1901.09207* (2019).
[27] Tianhao Zhang, Yueheng Li, Chen Wang, Guangming Xie, and Zongqing Lu. 2021. Fop: Factorizing optimal joint policy of maximum-entropy multi-agent reinforcement learning. In *International Conference on Machine Learning*. PMLR, 12491–12500.
[28] Stephan Zheng and Yisong Yue. 2018. Structured Exploration via Hierarchical Variational Policy Networks.
[29] Meng Zhou, Ziyu Liu, Pengwei Sui, Yixuan Li, and Yuk Ying Chung. 2020. Learning implicit credit assignment for cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems* 33 (2020), 11853–11864.