Coordination as Inference in Multi-Agent Reinforcement Learning

Zhiyuan Li[a,*], Lijun Wu[a], Kaile Su[b], Wei Wu[c,d], Yulin Jing[a], Tong Wu[a], Weiwei Duan[a], Xiaofeng Yue[a], Xiyi Tong[e], Yizhou Han[f]

[a]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
[b]School of Information and Communication Technology, Griffith University, Brisbane, Australia
[c]School of Computer Science and Engineering, Central South University, Changsha, China
[d]Xiangjiang Laboratory, Changsha, China
[e]Pittsburgh Institute, Sichuan University, Chengdu, China
[f]Glasgow International College, University of Glasgow, Glasgow, United Kingdom

Abstract

Despite the fact that the Centralized Training and Decentralized Execution (CTDE) paradigm has gained much attention due to its superior performance in the field of cooperative Multi-Agent Reinforcement Learning (MARL), fully decentralized policies learned using centralized training may fail due to the issue of Centralized-Decentralized Mismatch (CDM). In contrast to centralized learning, the cooperative model that most closely resembles the way humans cooperate in nature is fully decentralized, where the agents' policies are optimized independently, i.e. Independent Learning (IL). However, there are still two issues that need to be addressed before agents coordinate through IL: 1) how agents are aware of the presence of other agents, and 2) how to coordinate with other agents to improve joint policy under IL. In this paper, we propose an inference-based coordinated MARL method. Specifically, our approach consists of two parts. The first is individual intention modeling based on the conditional deep generative model, and the second is an attention mechanism and causal inference-based agent-level coordination. The proposed model was extensively experimented on a series of Multi-Agent MuJoCo and StarCraftII tasks. Results show that the coordination behavior among agents can be learned even without the CTDE paradigm compared to the state-of-the-art baselines including IPPO and HAPPO.

Keywords:
Multi-agent, Deep reinforcement learning, Partial observability, Non-stationary, Variational inference, Causal inference, Theory of mind

1. Introduction

One major challenge that has dominated the field of cooperative Multi-Agent Reinforcement Learning (MARL) for many years concerns how to learn coordinated behavior among multiple agents, especially in partially observable environments. This is due to the well-known non-stationarity of such domains, which is introduced by a group of agents interacting with each other in a shared environment and regarding the others as part of their environment. Previous work in this field usually adopts the paradigm of Centralized Training and Decentralized Execution (CTDE) to mitigate non-stationarity. Some promising approaches exploiting this paradigm include the value function decomposition methods (VDN [1], QMIX [2], QTRAN [3]), the communication-based methods (I2C [4], TMC [5]), and the coordination graph-based methods (DCG [6], CASEC [7]).

However, the value function decomposition method assumes that the optimal joint action represents the set of optimal action of each agent, and has a restrictive assumption on the decomposability of the joint Q-function. Unfortunately, these constraints lead to limited scalability. Broadcast communication is bandwidth-intensive and causes delays, while a large amount of redundant information makes it challenging to extract valuable cooperation information. Although extensive research has been carried out on succinct and robust communication recently, it still requires explicit communication channels with additional computational and memory overhead, which is difficult to deploy in decentralized control. The coordination graph-based method models the interactions between agents from the perspective of coordination graphs, but static and dense collaboration graphs may fail in dynamic environments because they induce intensive and invalid message passing. So far, however, there has been little discussion about how to learn dynamic and sparse coordination graphs.

In contrast, the cooperative model that most resembles human cooperation is a fully decentralized model in which the agents' policies are optimized independently, i.e., Independent Learning (IL). The IL paradigm, however, performs poorly even in simple cooperative tasks. One major issue that hinders agents from coordinating their own actions with those of others is that

*Corresponding author
Email addresses: zhiyuanli@std.uestc.edu.cn (Zhiyuan Li), wljuestc@sina.com (Lijun Wu), k.su@griffith.edu.au (Kaile Su), william.third.wu@gmail.com (Wei Wu), jingyulin@std.uestc.edu.cn (Yulin Jing), tongwu@std.uestc.edu.cn (Tong Wu), dwwuestc@163.com (Weiwei Duan), yxf@std.uestc.edu.cn (Xiaofeng Yue), wljuestc@Yahoo.com (Xiyi Tong), 2840640h@student.gla.ac.uk (Yizhou Han)

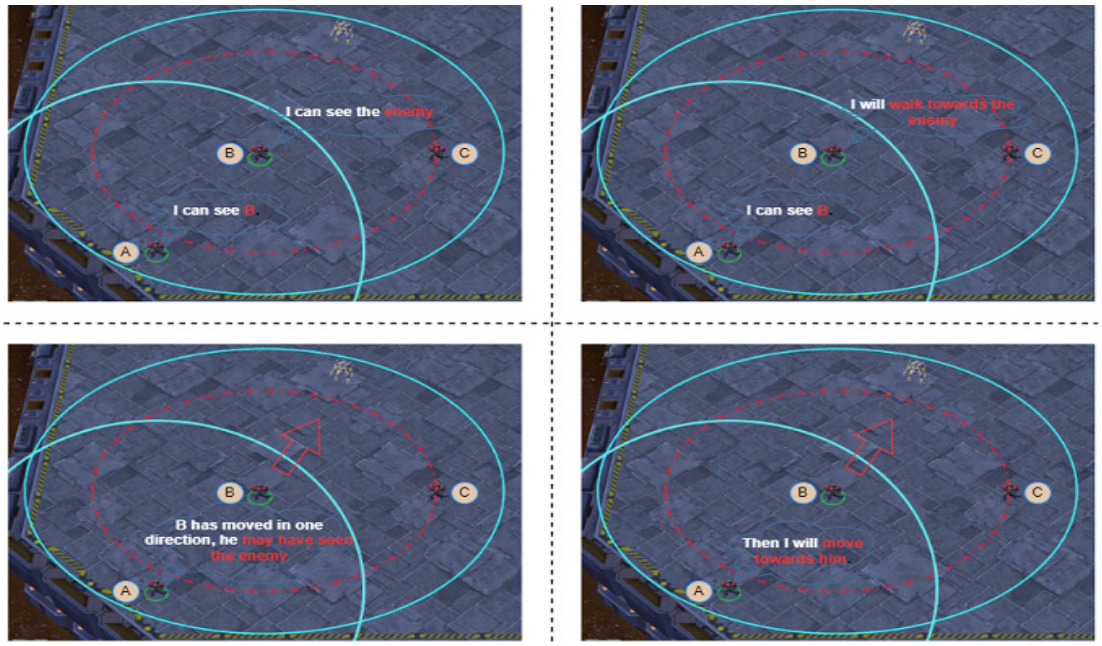


Figure 1: An example of StarCraftII Multi-Agent Challenge. Allies controlled by the agents need to cooperate to defeat the enemy. First, agents A and B observe the environment within their field of view. Agent B observes the enemy and therefore wants to attack it; however, as the enemy is out of its range, B makes a move toward it. Then, agent A observes B's movement and infers that B may have observed and intends to attack the enemy (i.e. intention of agent B). Therefore, agent A will move along the direction of B to try to cooperate with B in attacking the enemy.

they are unaware of the presence of other agents, which means that they are unable to adapt their policies to the actions of other agents. During interpersonal action coordination, people use their own motor system to simulate other people's actions in order to make predictions [8]. This allows them to evaluate others' intentions or motivations associated with higher-level, meta-representational thinking and to make decisions accordingly in social contexts, leading to the development of a cooperative relationship. In fact, humans are adept and instinctive at making inferences from others' external behavioral signals in order to deduce additional meanings behind the behavior. Imagine a team of players in a football match. Each player needs to predict his or her partner's movements in addition to reacting to each other's movements in order to cooperate correctly and effectively. In addition, the whole team must share common macro-goals and macro-intentions in order to effectively accomplish the joint action. This is the theory of mind (ToM) [9], also referred to as mentalizing or mental state reasoning. Such reasoning seeks to predict the relationships between external states of affairs and internal states of mind and to extract what information this contains about the distribution over individual observations.

To exploit this paradigm in cooperative multi-agent settings, we first propose a novel Deep Motor System (DMS) framework to encode the information (intention) contained in the observed actions, i.e., the fact that an agent decides to perform a specific action rather than another, which is crucial for learning efficient policies. Here, we take agent A in Fig. 1 as an example. Before acting, he needs to infer agent B's intention based on the movements of B within his field of view. Secondly, not every agent can provide useful information, and redundant information may even impair collaboration. Inspired by Causal Inference (CI), we believe that agents need to have beliefs about who to coordinate with; in other words, they are more inclined to cooperate with those who are potentially exerting more influence on their policies. Specifically, the intentions of other agents that lead to drastic changes in the agent's policy are considered relevant and are fed to the agent's policy network and value network. This is equivalent to increasing the mutual information between their intentions and filtering the intentions. CI can be viewed as the (hard) attention mechanism, widely used in many AI fields [10][11], as it selects a subset from other agents' intentions, entirely discarding the others. However, the hard-attention mechanism is non-differentiable. Therefore, it cannot learn the causal inference threshold directly through end-to-end back-propagation. Thus, we also introduce a soft-attention mechanism to calculate the importance distribution of other agents' intentions and generate "soft" filtered intentions. The filtered intentions can improve the agent's environment model as it may provide the agent with information about the parts of the state space that has not yet been accessed. This information can also be utilized to make agents explore the most promising region in the state space, reducing the exploration cost and significantly accelerating the convergence. Most importantly, the iterative reasoning and policy improvement among agents will continuously encourage cooperation and, even without an explicit Coalition Formation Algorithm, our method may lead to an ad-hoc phenomenon:

dynamic team composition. In addition, the iterative reasoning and structural relationships between non-local entities will enable agents to capture the high-level information required for solving complex problems. We evaluate the proposed method on benchmarks of StarCraftII and Multi-Agent MuJoCo against strong baselines such as IPPO [12] and HAPPO [13]. Results clearly demonstrate that independent learning can achieve coordinated behavior even in complex tasks without the CTDE paradigm, as well as without explicit communication or shared actions/observations.

2. RELATED WORK

Much of the current literature on MARL pays particular attention to achieving coordination among agents, which is a crucial and challenging task. Independent Q-Learning (IQL) [14] learns a separate Q-function for each agent to decentralize the policy of agents. The environment, however, becomes non-stationary since each agent continually changes its policy and considers other agents as being a part of its environment. As a result, most researchers utilized the CTDE paradigm, which assumes the availability of global information in the training phase and the distributed policy during execution. However, the CTDE paradigm may cause catastrophic miscoordination, i.e. Centralized-Decentralized Mismatch (CDM) [15]. Based on this paradigm, we classify recent studies into three categories.

The first approach determines each agent's contribution to the joint reward and then separates each agent's portion from it. This type of algorithm is called value function decomposition. VDN [1] assumes that the joint Q-function can be additively decomposed into N Q-functions for N agents. QMIX [2] guarantees monotonicity between the joint Q-function and the individual Q-functions via hypernetworks. QTRAN [3] further eliminates the constraint on non-negative weights in QMIX and provides a general decomposition for the joint Q-function. However, there remain several open questions about how to model sophisticated inter-agent coordination based on factorizations and how to learn those factorizations. Conversely, our method does not need any restrictive assumptions on the decomposability of the joint value function.

Second, many recent studies have shown that communication can improve agent coordination. [16] interprets cooperative communication as an optimal transport problem and derives some prior models as special cases. Zhang *et al.* [17] define the coordination problem from the perspective of game theory and propose a bi-level actor-critic learning method. However, they assume that agents make decisions in priority order. [18] proposes an inter-agent communication mechanism based on shared intentions. Assuming the shared parameters, [19] proposes to use autoencoding to learn communication. Prior to taking an action, SARNet [20] extracts the relevance of other agents' information and reason over received communications and past memories. Given the limited bandwidth, however, these broadcast-based methods may not work. Numerous studies have considered this constraint. For example, Wang *et al.* [21] prove that limited bandwidth requires low-entropy messages, and propose IMAC to learn efficient communication protocols and scheduling. To reduce information redundancy, I2C [4] enables agents to learn a priori of communication through causal inference. TMC [5] provides significantly reduced communication overhead and better transmission loss robustness. [22] proposes an algorithm that synthesizes a control policy that combines a programmatic communication policy for decentralized control of multi-agent systems, which significantly reduces the amount of communication. Our approach differs from the communication-based approach in that we coordinate via inference rather than communication.

The third method models coordination behavior with a coordination graph. The multi-agent environment is represented as a graph where each agent is a vertex with the encoding of the agent's local observation as its feature, and there is an edge between a vertex and each of its neighbors. Ruan *et al.* [23] introduce a graph generator and a graph-based coordination policy to dynamically represent the underlying decision dependency structure. DCG [6] decomposes the joint value function into payoffs between pairs of agents according to the coordination graph. However, static and dense collaboration graphs in DCG may fail in dynamic environments and induce intensive and inefficient message passing. In contrast to DCG, [7] focuses on learning the dynamic sparse coordination graph, using the variance of pairwise payoff functions as an indicator to select edges. Unlike such methods, we model coordinated behavior implicitly via inference rather than coordination graphs.

Several studies do not belong to the above three categories of methods (for example, Jaques *et al.* [24]; Witt *et al.* [25]; Zhang *et al.* [26]), however, some of them need to modify the reward function or only consider the coordination between neighbors. In contrast to the CTDE paradigm, this study addresses the issue of CDM via the IL paradigm and proposes an implicit coordination model to learn coordinated behavior and stabilize independent learning. The closest work to ours is Jaques *et al.* [24], Ding *et al.* [4], Strouse *et al.* [27], and Tian *et al.* [28]. Jaques *et al.* [24] and Ding *et al.* [4] design agents that take into account the causality between self-other agents, but they either need to modify the reward function or still require an explicit communication channel. Strouse *et al.* [27] also models the intention of agents, while it still belongs to the CTDE paradigm and needs reward shaping. Our work is partly in line with Tian *et al.* [28], where the authors propose PBL for learning collaborative behavior without any explicit communication. One difference between our work and Tian *et al.*'s [28] is that we do not require private information of other agents. Moreover, we do not need any auxiliary rewards or modifications. Instead, we guide the policy evaluation and policy improvement through intention modeling and update the intention according to the observed historical action trajectory. The resulting coordination mechanism is consistent during training and execution.

3. PRELIMINARIES

In this section, we first introduce the cooperative MARL problem. Then we review the conditional generative model and finally introduce one of the SOTA algorithms.

3.1. Cooperative MARL Problem Formulation

We consider a cooperative multi-agent system that is often modeled as Decentralized Partially Observable Markov Decision Processes (Dec-POMDPs). The Dec-POMDPs is defined as $G = \langle \mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{N} = \{1, \ldots, n\}$ is a set of agents, $s \in \mathcal{S}$ denotes the state of the environment, $\mathcal{A} = \prod_{i=1}^{n} A^i$ is the product of the agents' action spaces, namely the joint action space, and $\mathcal{O} = \prod_{i=1}^{n} O^i$ is the joint observation space. Each agent $i \in \mathcal{N}$, at time step $t \in \mathbb{N}$ first draws an individual observation $o_t^i \in O^i$ ($o_t = \left\{o_t^1, \ldots, o_t^n\right\}$ is the joint observation of agents), and then takes an action a_t^i according to its policy $\pi^i(\cdot|o_t^i)$ ($\pi(\cdot|o_t) = \prod_{i=1}^{n} \pi^i\left(\cdot|o_t^i\right)$ is the joint policy). With the joint action of agents $a_t = \left\{a_t^1, \ldots, a_t^n\right\}$, the environment moves to a state s' with probability $\mathcal{P}(s'|s, a_t)$. Each agent receives a joint reward $\mathcal{R}(s_t, a_t)$ and observes o_{t+1}, whose probability distribution is $\mathcal{P}(\cdot|o_t, a_t)$. In an episode containing T steps, the objective of each agent i is to maximize the cumulative discounted return $\mathcal{R}^\gamma \triangleq \sum_{t=0}^{T} \gamma^t \mathcal{R}(s_t, a_t)$, where $\gamma \in [0, 1)$ is a discounted factor.

3.2. Conditional Generative Model

A generative model with certain types of latent variables, such as Gaussian latent variables, has the form of $p(x) = \sum_z p(x|z)\, p(z)$, known as Variational Auto-Encoder (VAE). The VAE contains two distributions: the prior distribution $p_\theta(z)$ is used to construct a set of latent variable z, and the generative distribution $p_\theta(x|z)$, conditional on z, is used to generate the data x.

Due to intractable posterior inference, the parameter estimation of VAE is typically challenging. However, the parameters of VAE can be estimated efficiently with the evidence lower bound (ELBO) which is used as a surrogate objective function w.r.t. ϕ:

$$ELBO_z = \mathbb{E}_{q_\phi(z|x)}\left[log\ p_\theta(x|z)\right] - KL\left[q_\phi(z|x)\,\|p_\theta(z)\right] \quad (1)$$

Conditional VAE (CVAE) extends the standard VAE framework with auxiliary covariates. CVAE is composed of multiple MLPs, such as encoder network $q_\phi(z|x, y)$, conditional prior network $p_\phi(z|x)$, and decoder network $p_\theta(y|x, z)$. The ELBO for the CVAE can be obtained by conditioning the probabilities in Eq. 1 with the covariates.

3.3. Heterogeneous-Agent Proximal Policy Optimisation

HAPPO is currently one of the SOTA algorithms that attain theoretically-justified monotonical improvement property. The key is the multi-agent advantage decomposition lemma [29]. During training, HAPPO draws a random permutation of agents $i_{1:n}$, and then following the order in the permutation, every agent i_m picks π^{i_m} that maximizes the objective of

$$\begin{aligned} L_{PG}^{i_m} =& \mathbb{E}_{s\sim\rho_{\pi_{old}}, a^{i_{1:m-1}}\sim\pi^{i_{1:m-1}}, a^{i_m}\sim\pi_{old}^{i_m}}[min(\\ & \frac{\pi^{i_m}(a^{i_m}|s)}{\pi_{old}^{i_m}(a^{i_m}|s)} M^{i_{1:m}}(s,a), \\ & clip(\frac{\pi^{i_m}(a^{i_m}|s)}{\pi_{old}^{i_m}(a^{i_m}|s)}, 1\pm\epsilon) M^{i_{1:m}}(s,a))], \end{aligned} \quad (2)$$

where $M_{i_{1:m}} = \frac{\pi^{i_{1:m-1}}(a^{i_{1:m-1}}|s)}{\pi_{old}^{i_{1:m-1}}(a^{i_{1:m-1}}|s)} \hat{A}(s,a)$, and $\hat{A}(s,a)$ is an estimate of advantage function with generalized advantage estimation (GAE) [30] as the robust estimator.

4. MULTI-AGENT COORDINATION VIA INFERENCE

In an unknown cooperative setting, each agent is unaware of the existence of other entities and thus treats them as a part of its environment. Moreover, due to the ignorance of the intentions of other agents, each agent is not capable of coordinating with the others, which makes it challenging for agents to learn coordination policies in a non-stationary environment. Therefore, DMS in Fig. 2 aims to learn the representation of the other agents' intentions, which consists of an encoder network, a prior network, and a decoder network. Agent-level coordination measures the correlation between agents via causal inference or attention from the perspective of the individual.

4.1. Deep Motor System

In partially observable environments, each agent selects an action conditioned on its individual observation, and cannot access to either the global state or joint observation. To mitigate partial observability, we integrate the individual observations with the intentions of other agents to (partially) reconstruct the global state or joint observation as the intentions contain information about their observation distributions. Thus, prior to learning coordinated relationships, we consider the following question:

"How can we accurately model the intention of agents?"

Human collaborators are expertise in social perception, defined as the ability to encode another person's mental states based on basic behavioral signals. In addition, when people comprehend others' actions, there is another mechanism that relies heavily on the perceivers' own motor system: in order to understand others' action, the perceivers must first observe the action, then form a mental imitation of it, and make a comparison between the observed and imitated action through the mirror neural system [31].

Based on the above theory, DMS is a deep conditional generative model for intention modeling based on historical action trajectory. At the end of an episode containing T steps, each agent stores the observed historical action trajectory of other agents, denoted by $\mathbf{A}$. Given $T_{sep} < T$, we divide $\mathbf{A}$ into past actions $\mathbf{A}_P = \mathbf{A}^{1:T_{sep}}$ and future actions[1] $\mathbf{A}_F = \mathbf{A}^{(T_{sep}+1):T}$. The

[1]The future here is relative to the past, not real future actions.

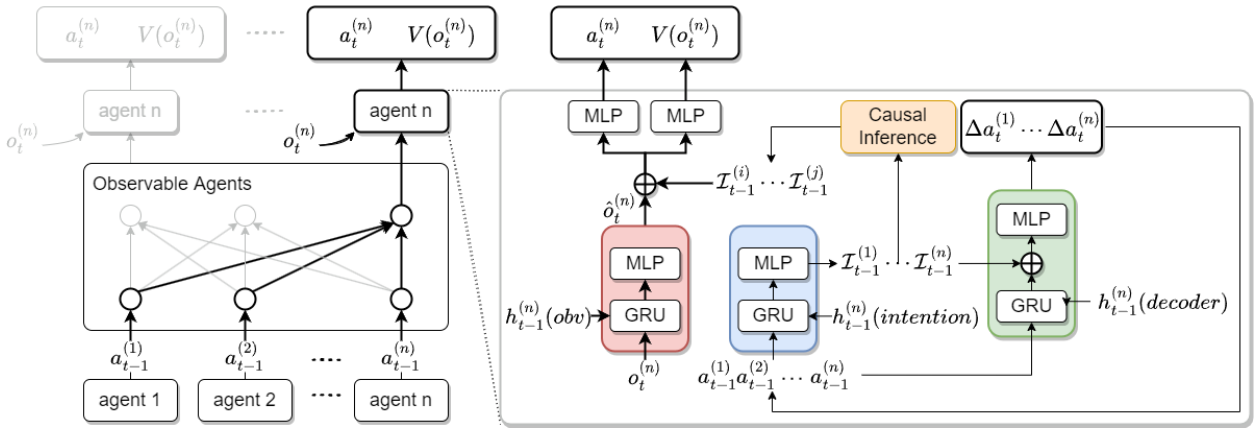


Figure 2: The architecture of DMS. It consists of a policy/value network (red module), an encoder/prior (blue module), a decoder (green module), and a causal inference module (orange). Each network involves a Gated Recurrent Unit (GRU) and a Multilayer Perceptron (MLP). At each time step, each agent receives local observation as well as the actions of the other agents within its field of view during the last time step. The policy/value network takes the local observation $o_t^{(n)}$ and historical observation information $h_{t-1}^{(n)}(obv)$ as inputs and produces an learned representation of the environment $\hat{o}_t^{(n)}$. The encoder/prior accepts the observed actions $a_{t-1}^{(1)} \cdots a_{t-1}^{(n)}$ of other agents during the last time step and historical intention information $h_{t-1}^{(n)}(intention)$ and then produces other agents' intention $\mathcal{I}_{t-1}^{(1)} \cdots \mathcal{I}_{t-1}^{(n)}$. Upon generating the intentions, the causal inference module quantifies the causal effects between other agents and produces the filtered intentions $\mathcal{I}_{t-1}^{(i)} \cdots \mathcal{I}_{t-1}^{(j)}$ which are then combined with $\hat{o}_t^{(n)}$ to generate the next action and value. The intentions combined with $h_{t-1}^{(n)}(decoder)$ enter the decoder to imitate $\Delta a_t = a_t - a_{t-1}$.

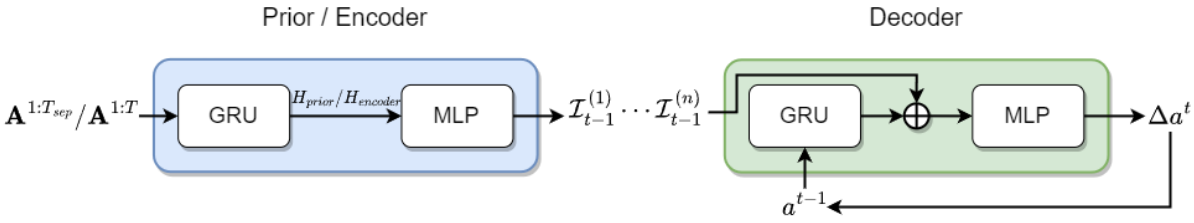


Figure 3: Details of DMS.

task is to generate other agents' intentions and restore $\mathbf{A}_F$ based on $\mathbf{A}_P$, that is, mental imitation. During training, we use the encoder given $\mathbf{A}_P$ and $\mathbf{A}_F$ to infer the latent variable $\mathcal{I}$ (i.e. intention) and optimize ELBO with Adam [32]. During execution, we generate the intention using the prior network given the historical action trajectory. Fig. 3 illustrates DMS. We describe the details in the following subsections.

4.1.1. Encoder

Humans are capable of temporal integration, that is, the process and ability to construct and integrate information across time into a coherent whole, allowing one to comprehend and predict occurrences over time. Hence, we concatenate $\mathbf{A}_P$ and $\mathbf{A}_F$ along the time dimension, use Gated Recurrent Unit (GRU) [33] to integrate the feature over the whole historical action trajectory and perform feature regularization using layer normalization [34]:

$$H_{Encoder} = LN(GRU([\mathbf{A}_P, \mathbf{A}_F])) \tag{3}$$

We assume that the latent variable $\mathcal{I} \sim q(\mathcal{I}|\mathbf{A}_P, \mathbf{A}_F)$ follows a multi-variate Gaussian distribution $N(\mu, \mathbf{I}\sigma)$, where the μ and σ are computed by Fully Connected Layer (FCL). We sample $\mathcal{I}$ with the reparameterization trick [35].

4.1.2. Prior

During execution, the agent can only model the intention $\mathcal{I}$ from the recorded historical action trajectory:

$$H_{Prior} = LN(GRU(\mathbf{A}_P)) \tag{4}$$

Thus, the prior $p(\mathcal{I}|\mathbf{A}_P)$ represents the conditional distribution of $\mathcal{I}$ given only $\mathbf{A}_P$. We make the prior and the encoder share the same weight so that the only difference is input. Optimizing Eq. 1 minimizes the Kullback-Leibler (KL) divergence between information extracted from the prior and the encoder.

4.1.3. Decoder

The decoder forms a mental imitation of actions and com-

pares the observed and imitated actions. Hence, agents can adapt their own actions in accordance to these mental imitations, which facilitates fast and accurate interpersonal coordination. Specifically, the decoder $p(\mathbf{A}_f|\mathcal{I}, \mathbf{A}_P)$ imitates the future action trajectory conditioned on the intention $\mathcal{I}$ and the past action trajectory $\mathbf{A}_P$. Here, we adopt the function that predicts the difference between the next action and the current action, i.e. $a_{t+1} - a_t$, rather than the next action value a_{t+1}, to reduce the model bias in the early stage of learning.

4.2. Agent-level Coordination

On the one hand, as indicated by DMS, social interaction requires neural representation of others' mental states and corresponding beliefs in the intentions of partners. Moreover, each agent of the interaction has its own DMS, and these systems are continuously adjusted. Coordinated adjustments of DMSs and behavioral policies, however, can only occur if the neural processes of the agents are synchronized, as inter-brain synchronization is a crucial and inevitable mechanism for interpersonal action coordination and social interaction. Therefore, we argue that in addition to understanding the behavior of others, perceiving agents should also successfully match their own policy with their partners' by being aware of the causal effects of others' intentions on their policy. Intuitively, an agent is more likely to collaborate with entities that are more likely to exert significant influence on its policy, hoping to get information about their propensities for acting and how to react cooperatively. Hence, the causal effects of other agents can be regarded as the necessity of making decisions conditioned on the intention of other agents.

On the other hand, assuming that we have obtained the intentions of all partners, it is justified to ask here: why is inferring the causality between the intentions needed instead of directly integrating them all if the perceiver is already tracking and predicting their partner's intentions? In our view, simple intention integration may lead to information redundancy, and impair the learning process, whereas causal inference enables agents to capture high-order relationships, and even form dynamic ad-hoc teams with common goals and intentions.

In this paper, we quantify the causal effects between the agents via the KL divergence between the policy distributions conditioned on different intentions:

$$C_{ij} = KL\left[\pi^i\left(a^i \mid o^i, \mathcal{I}^{-i}\right) \| \pi^i\left(a^i \mid o^i, \mathcal{I}^{-ij}\right)\right], \tag{5}$$

where $\mathcal{I}^{-i}$ denotes the intentions of other agents except for agent i, and $\mathcal{I}^{-ij}$ denotes the intentions of other agents except for agent i and j. KL divergence is used to measure how different two conditional probability distributions are. The value of C_{ij} represents the degree of change in the policy distribution of agent i taking into account the intention of agent j, as well as the correlation between agents i and j. If C_{ij} is greater than a threshold δ, then there is a strong correlation between the two agents, and agent j will be added to the coordination set of agent i; otherwise, it is a weak correlation. The threshold δ is a hyperparameter set to 0.6, and the performance of different threshold δ is analyzed in the experiment.

Causal inference selects a subset from other agents' intentions, which entirely discards the others. However, the threshold δ is non-differentiable and it cannot be learned through backpropagation. Thus, we also introduce soft-attention mechanism to approximate inter-brain synchronization by a weighted average of each agent's intention. The attention enables each agent to process the intentions of other agents differently depending on their learned importance distribution:

$$e_{ij} = softmax(\mathcal{I}^i \cdot \mathcal{I}^j), \tag{6}$$

where e_{ij} is the importance weight for intention $\mathcal{I}^j$.

Each agent makes policy improvement based on the intentions of some of the other agents, and the resulting new intention is also modeled by the others. As a result, the iterative reasoning and policy improvement among agents will lead to infinite nested belief [36] and agent-level coordination.

4.3. Long Short-Term Intention

A trick in our work is the use of long short-term intention. The agents may perform actions that do not match the long-term intention due to the exploration, thus interfering with intention modeling. Moreover, in competitive settings, the competitors may take deceptive actions in a short period of time (contrary to long-term intention). Hence, we propose the short-term intention network to eliminate distractive actions and to learn robust intention.

The short-term intention network shares the parameters with the prior network in DMS and differs only in the input of the GRU (Eq. 4). With the shared weight, the same GRU sees inputs with different lengths. Specifically, the prior takes as input the entire historical action trajectory, while the short-term intention network uses the action trajectory of length l, where l is set to 4. The performance of different l is analyzed in the experiment. We calculate the KL divergence between the intentions modeled by the prior and the short-term intention network, respectively. If the value of KL divergence is greater than a threshold K, the l actions will be masked; otherwise kept.

4.4. Action Masking

In a multi-agent environment, each unit is controlled by an independent agent that conditions only on local observations restricted to a limited field of view centered on that unit (e.g., StarCraftII Multi-Agent Challenge [37]). Thus, it is impossible for an agent to observe the actions of all others at every moment, and there will be defaults in the action trajectories observed by the agent. Modeling intentions for these defaults is meaningless and can hinder policy learning. In order to prevent DMS from modeling the defaults, our suggestion is to simply use an agent-specific constant vector, i.e., a zero vector with the agent's ID, as the input to DMS. We call this technique Action Masking.

4.5. Training

4.5.1. HAPPO

Update V-value network of agent i_m by following formula:

$$L_{value}^{i_m}(\vartheta) = \frac{1}{BT}\sum_{b=1}^{B}\sum_{t=0}^{T-1}[\mathcal{R}(s_t,a_t)+\gamma V_{\vartheta}(o_{t+1},\mathcal{I}_{t+1}^{-i})- V_{\vartheta}(o_t,\mathcal{I}_t^{-i})]^2, \tag{7}$$

where B is the batch size.

Update[2] actor i_m by Eq. 2.

4.5.2. Deep Motor System

Each agent collects T timesteps of the observed historical actions and stores them into buffer $\mathcal{B}$. During training, we sample a minibatch of historical actions from $\mathcal{B}$ and divide it into $\mathbf{A}_P$ and $\mathbf{A}_F$. Then $\mathbf{A}_P$ and $\mathbf{A}_F$ are concatenated as the input of the encoder to obtain $q_\phi(\mathcal{I}|\mathbf{A}_P,\mathbf{A}_F)$, from which the intention $\mathcal{I}$ is sampled, while the prior only takes $\mathbf{A}_P$ as input to produce $p_\phi(\mathcal{I}|\mathbf{A}_P)$. The decoder $p(\mathbf{A}_F|\mathcal{I},\mathbf{A}_P)$ reconstructs $\mathbf{A}_F$ conditioned on intention $\mathcal{I}$ and $\mathbf{A}_P$. $p(\mathbf{A}_F|\mathcal{I},\mathbf{A}_P)$ follows a Gaussian distribution or a Categorical distribution respectively according to whether the action space is continuous or not. We then optimize the ELBO:

$$ELBO_{\mathcal{I}} = \mathbb{E}_{q(\mathcal{I}|\mathbf{A}_P,\mathbf{A}_F)}[log\ p(\mathbf{A}_F|\mathcal{I},\mathbf{A}_P)]- KL[q(\mathcal{I}|\mathbf{A}_P,\mathbf{A}_F)\|p(\mathcal{I}|\mathbf{A}_P)] \tag{8}$$

We approximate the expectation with the sample mean and if the decoder follows a Gaussian distribution:

$$\mathbb{E}_{q(\mathcal{I}|\mathbf{A}_P,\mathbf{A}_F)}[log\ p(\mathbf{A}_F|\mathcal{I},\mathbf{A}_P)] \approx \frac{1}{Z}\sum_{t=T_{sep}+1}^{T}\sum_{i=1}^{n} [\frac{m}{2}log(2\pi\hat{\sigma}^2)+\frac{\sum_{j=1}^{m}(\hat{a}_t^i-a_t^i)^2}{2\hat{\sigma}^2}], \tag{9}$$

or a Categorical distribution:

$$\mathbb{E}_{q(\mathcal{I}|\mathbf{A}_P,\mathbf{A}_F)}[log\ p(\mathbf{A}_F|\mathcal{I},\mathbf{A}_P)] \approx \frac{1}{Z}\sum_{t=T_{sep}+1}^{T}\sum_{i=1}^{n} [a_t^i log(\hat{\theta}_t^{a_t^i})], \tag{10}$$

where $\hat{\theta}_t^{a_t}$ is the probability of reconstructing a_t at step t, Z is the normalization term, and m is the dimension of the action space. Since $q(\mathcal{I}|\mathbf{A}_P,\mathbf{A}_F)$ and $p(\mathcal{I}|\mathbf{A}_P)$ follow Gaussian distributions, the KL divergence term can be computed analytically:

$$KL[q(\mathcal{I}|\mathbf{A}_P,\mathbf{A}_F)\|p(\mathcal{I}|\mathbf{A}_P)] = log(\frac{\sigma_{Encoder}}{\sigma_{Prior}})+ \frac{\sigma_{Encoder}^2+(\mu_{Encoder}-\mu_{Prior})^2}{2\sigma_{Prior}^2}-\frac{1}{2} \tag{11}$$

[2]The state s in Eq. 2 is replaced with $(o,\mathcal{I}^{-i_m})$.

Since DMS being updated is also used in calculating the policy gradient and the TD error, performing multiple steps of optimization on $L_{PG}^{i_m}$, $L_{value}^{i_m}$, and ELBO encourages DMS to update towards the objective of maximizing the cumulative discounted return and ELBO. However, policy and value networks may diverge due to dramatic changes in DMS. Our solution is to use "soft" updates, that is, updating DMS via Polyak averaging [38]: $\phi_{new} = \rho\phi_{new} + (1-\rho)\phi_{old}$ with $\rho \ll 1$. This means that DMS is constrained to change slowly, which greatly improves the stability of learning. Furthermore, to encourage agents to reconstruct the joint observation, we propose an optional feature-wised reconstruction loss which incentives agents to obtain global state information from others' intentions:

$$\mathbb{E}_{o,\mathcal{I}^{-i_m},s}[(\phi(o,\mathcal{I}^{-i_m})-\phi(s)^2] \tag{12}$$

where ϕ indicates the feature representations of $(o,\mathcal{I}^{-i_m})$ or s.

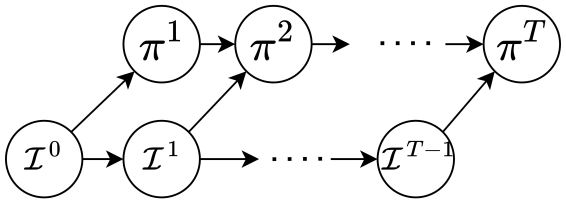


Figure 4: Lag between update of intention and policy.

We also apply several training tricks to further stabilize the learning: (1) Delayed Updates: Biased intention approximation may lead to poor Critic and Actor updates, which is particularly problematic, as the wrong intention will lead to inaccurate policy evaluation, which will cause the policy improvement in the wrong direction, and biased intention will also directly affect the policy gradient. Hence, we introduce the delayed update proposed in TD3 [39], where the policy and value network are updated at a lower frequency than DMS, to first minimize intention loss before introducing an actor-critic update. Specifically, we only update the policy and value networks after a fixed number of updates d to DMS. Besides, after each updating step, Actor-Critic iterates to the latest parameters, while DMS models the intention of the other agents at the previous step, which leads to a lag (see Fig. 4). From the perspective of Generative Adversarial Network (GAN) [40], we can consider DMS as a discriminator and Actor-Critic as a generator. Hence, we alternate between optimizing DMS for d steps and Actor-Critic for one step. This will cause DMS to remain near its optimal solution (accurately modeling the intention) as long as the Actor-Critic changes slowly enough. (2) Pre-training: In the early stages of training, intention updates are unstable since the agent's policy is quite random. Therefore, We pre-train the first 100 episodes, i.e., learning without intention in the early stage. Algorithm 1 represents the pseudo-code for training.

Algorithm 1 Coordination as Inference

Input: Step-size α, batch-size B, number of agents n, episodes K, steps per episode T, pre-training episodes P, delayed updates d, Polyak averaging ρ.

Initialize: Actor networks $\{\theta_0^i, \forall i \in \mathcal{N}\}$, Value networks $\{\vartheta_0^i, \forall i \in \mathcal{N}\}$, Encoders $\{\phi_0^i, \forall i \in \mathcal{N}\}$, Priors $\{\varphi_0^i, \forall i \in \mathcal{N}\}$, Decoders $\{\psi_0^i, \forall i \in \mathcal{N}\}$, Replay buffer $\mathcal{B}$, $\{a_{-1}^{-i} = [defaults], \forall i \in \mathcal{N}\}$.

for $k = 0, 1, \ldots, K-1$ **do**
  for $t = 0, 1, \ldots, T-1$ **do**
    for $i = 0, 1, \ldots, n-1$ **do**
      Collect individual observation o_t^i from environments, and last step observed actions a_{t-1}^{-i}.
      if $k < P$ **then**
        Generate observation feature $\hat{o}_t^i$.
      else
        Generate $\mathcal{I}^{-i}$ by feeding a_{t-1}^{-i} to ϕ_k^i.
        Compute C_{ij} or e_{ij} $\forall j \in \mathcal{N} \backslash \{i\}$ and get the coordination set of agent i.
        Get filtered intention $\mathcal{I}_f^{-i}$ according to the coordination set.
        Generate observation feature $\hat{o}_t^i$ by concatenating $[o_t^i, \mathcal{I}_f^{-i}]$.
      end if
      Input $\hat{o}_t^i$ to θ_k^i and infer a_t^i.
    end for
    Execute joint actions $a_t^0, \ldots, a_t^{n-1}$ in environments and collect the reward $\mathcal{R}(o_t, a_t)$ and the observed actions $\{a_t^{-i}, \forall i \in \mathcal{N}\}$.
    Push transitions $\{(o_t^i, a_t^i, o_{t+1}^i, \mathcal{R}(o_t, a_t), a_t^{-i}), \forall i \in \mathcal{N}, t \in T\}$ into $\mathcal{B}$.
  end for
  Sample a minibatch of B steps from $\mathcal{B}$.
  Draw a random permutation of agents $i_{0:n-1}$.
  for $i = i_0, \ldots, i_{n-1}$ **do**
    if $k > P$ **then**
      // Update DMS.
      Input $[\mathbf{A}_P, \mathbf{A}_F]$ and $\mathbf{A}_P$ to ϕ_k^i and φ_k^i, respectively and infer $q(\mathcal{I}|\mathbf{A}_P, \mathbf{A}_F)$, $p(\mathcal{I}|\mathbf{A}_P)$.
      Sample intention $\mathcal{I}^{-i}$ from $q(\mathcal{I}|\mathbf{A}_P, \mathbf{A}_F)$.
      Input $\mathcal{I}^{-i}$ and $\mathbf{A}_P$ to ψ_k^i to reconstruct $\hat{\mathbf{A}}_F$.
      Calculate ELBO with Eq. 8.
      Update the encoder and decoder:
      $\phi_{k+1} = \rho\phi_{k+1} + (1-\rho)\phi_k$
      $\psi_{k+1} = \rho\psi_{k+1} + (1-\rho)\psi_k$
    end if
    // Update the actor and value networks.
    Generate $V_{\vartheta_k^i}(\hat{o}_t^i)$.
    Calculate $L_{value}(\vartheta_k^i)$ with Eq. 7.
    Compute individual advantage function $\hat{A}^i$ based on $V_{\vartheta_k^i}(\hat{o}_t^i)$ with GAE.
    Calculate $L_{actor}(\theta_k^i)$ with Eq. 2.
    if k mod d **then**
      Update the actor and value networks with Adam.
    end if
  end for
end for

5. EXPERIMENTS AND RESULTS

In this section, we evaluate our work against several state-of-the-art baselines on two most common benchmarks: StarCraftII Multi-Agent Challenge (SMAC) and Multi-Agent MuJoCo. We then examine the aspects of this work by providing further ablation studies of the components of the model.

5.1. Environmental Setting

5.1.1. StarCraftII Multi-Agent Challenge (SMAC).

SMAC is based on the popular real-time strategy game StarCraft II and focuses on micromanagement challenges where each unit is controlled by an independent agent that must act based on local observations. SMAC consists of a set of StarCraft II micro scenarios (e.g. 3s5z, 8m_vs_9m, MMM2) which aim to evaluate how well independent agents are able to learn coordination to solve complex tasks (Fig. 5(a)-5(b). Each scenario is a confrontation between two armies of units.

5.1.2. Multi-Agent MuJoCo.

Multi-Agent MuJoCo is a novel benchmark for decentralized cooperative continuous multi-agent robotic control. Multi-Agent MuJoCo decomposes single robots into individual segments controlled by different agents (Fig. 5(c)-5(d). We set the maximum observation distance k to 1 and configure all multi-agent MuJoCo environments using the default settings.

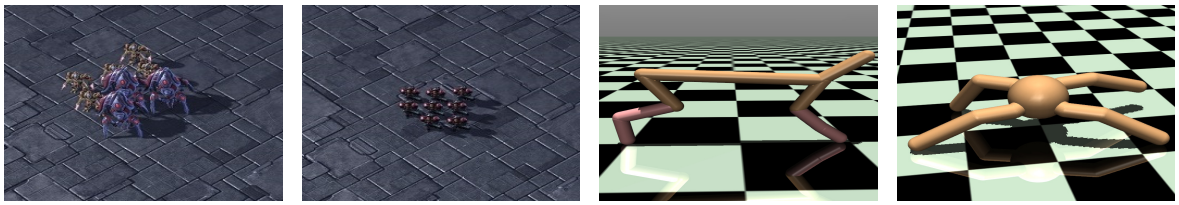
5.1.3. Baselines.

We compare our work with several state-of-the-art baselines as follows. IPPO and HAPPO have been proposed to learn monotonically improving policies in multi-agent settings.

- IPPO: Independent PPO (IPPO), a multi-agent variant of proximal policy optimization [41] for decentralized training in multi-agent systems. Each agent receives its local history of observations and updates the parameters of networks.
- HAPPO: IPPO lacks the essential theoretical property of trust region learning, which is the monotonic improvement guarantee. Conversely, Heterogeneous-Agent Proximal Policy Optimisation (HAPPO) fully leverages *Multi-Agent Advantage Decomposition Theorem* and the *sequential policy update scheme* to implement multi-agent trust-region learning with monotonic improvement guarantee. For fair comparison, we denote HAPPO-il as an independent learning variant of HAPPO.

We re-implement the baselines following their original paper to ensure their best performance. All hyperparameter settings and implementation details can be found in Table 1-4.

All the experiments are performed in a computer equipped with an Intel® Xeon™ Gold 5218R CPU @ 2.1GHz and an Nvidia GeForce RTX 3070 GPU.



(a) 3s5z (b) 8m vs 9m (c) Halfcheetah (d) Ant

Figure 5: Demonstrations of the SMAC and the Multi-Agent MuJoCo environments.

Table 1: Common hyperparameters in the SMAC domain

hyperparameters	value	hyperparameters	value
rollout threads	20	training threads	32
episode length	160	num mini-batch	1
gain	0.01	optim eps	1e-5
hidden size	64	stacked-frames	1
ppo epoch	5		

Table 2: Different hyperparameters in the SMAC domain

Algorithms	IPPO	HAPPO-il	HAPPO	Our
actor lr	5e-4	5e-4	5e-4	1e-5
critic lr	5e-4	5e-4	5e-4	1e-4
intention lr	/	/	/	1e-3
gamma	0.99	0.95	0.95	0.99
max grad norm	10	10	10	0.5
ppo clip	0.2	0.3	0.3	0.2
actor update freq	/	/	/	2
critic update freq	/	/	/	2
δ	/	/	/	0.6
K	/	/	/	1e4
l	/	/	/	4
pre-training	/	/	/	100

Table 3: Common hyperparameters in the Multi-Agent MuJoCo domain

hyperparameters	value	hyperparameters	value
rollout threads	4	training threads	8
episode length	1000	num mini-batch	1
gain	0.01	optim eps	1e-5
hidden size	64	stacked-frames	1
ppo epoch	5	gamma	0.99

Table 4: Different hyperparameters in the Multi-Agent MuJoCo domain

Algorithms	IPPO	HAPPO-il	HAPPO	Our
actor lr	5e-6	5e-6	5e-6	1e-6
critic lr	5e-3	5e-3	5e-3	1e-4
intention lr	/	/	/	1e-3
max grad norm	10	10	10	0.5
ppo clip	0.2	0.3	0.3	0.2
actor update freq	/	/	/	2
critic update freq	/	/	/	2
δ	/	/	/	0.6
K	/	/	/	1e4
l	/	/	/	4
pre-training	/	/	/	100

5.2. Parameter Tuning

We first study the impact of the coordination threshold (Section 4.2), controlled by the hyperparameter, δ. We fine-tune the coordination threshold by selecting the best model in the two domains. We consider three δ values: 0.2, 0.6, and 0.8. Table 5 demonstrates that the model with δ being 0.6 achieves the highest winning rate in the SMAC domain. When a lower value is used, the agents tend to utilize the intentions of all other agents. Although the agent performs well at the beginning of training, information redundancy makes the agent's performance to become inconsistent. Conversely, when using higher δ values, the model may fail to coordinate with any other agents due to the high threshold for cooperation and the results are similar to those of HAPPO-il (see Table 5 and Table 6). As shown in Fig. 6, similar conclusions can be drawn in the Multi-Agent MuJoCo Manyagent Swimmer scenario.

We then examine the effect of long short-term intention threshold K and length of short-term action trajectory l (Section 4.3). With small K such as 1e2 and long trajectory length such as 6, short-term action trajectories have a greater chance of being discarded, and thus DMS's learning speed is slowed. However, with such (K, l), the performance is more stable, as demonstrated by the smaller standard deviation in the training curves. We also observe that large K 1e6 and small l 2 can result in inconsistent performance. Fig. 7 demonstrates that $(1e4, 4)$ achieves higher reward than other values, as short-term intentions are better captured and anomalous intentions are effectively eliminated. Besides, the results show the necessity of long short-term intention module which eliminates distractive actions and helps the agents obtain valuable information and learn better policies.

We fix these parameters in the remaining part of our test.

5.3. Main Results

Here we report the experimental results from the environmental setting aforementioned in Section 5.1.

Table 5: The effect of coordination threshold on DMS's win rate and standard deviation in SMAC.

Map	0.2	0.6	0.8
3s5z	$96.9_{1.5}$	$\mathbf{97.5_{2.2}}$	$89.8_{2.7}$
8m vs 9m	$84.4_{3.8}$	$\mathbf{82.6_{6.1}}$	$77.5_{9.7}$
MMM2	$55.9_{7.4}$	$\mathbf{62.1_{8.8}}$	$43.2_{12.5}$

Figure 6: The effect of coordination threshold on DMS's performance in Manyagent Swimmer scenario.

5.3.1. SMAC Results

In this section, we compare the proposed approach with IPPO, HAPPO, and HAPPO-il on a number of SMAC maps (see Fig. 5(a)-5(b)). We compute the median winning rate over 10 test games after each training phase for 6 training seeds in Table 6. In each scenario, our model and each baseline are trained over steps shown in the Step column of the Table. As shown in Table 6, the proposed method outperforms IPPO and HAPPO-il in almost all scenarios, indicating its superior performance over such independent learning algorithms, while HAPPO outperforms our method by a small margin.

Table 6: Performance evaluations of winning rate and standard deviation on the SMAC benchmark.

Map	IPPO	HAPPO-il	HAPPO	Our	Steps
3s5z	$98.4_{3.1}$	$90.0_{3.5}$	$99.1_{1.8}$	$97.5_{2.2}$	1e7
8m vs 9m	$81.8_{7.3}$	$72.5_{4.4}$	$87.3_{2.6}$	$82.6_{6.1}$	1e7
MMM2	$51.8_{10.1}$	$39.2_{9.8}$	$63.3_{4.0}$	$62.1_{8.8}$	1e7

5.3.2. Multi-Agent MuJoCo Results

Multi-Agent MuJoCo is a benchmark for continuous multi-agent robotic control which is based on OpenAI's MuJoCo Gym environments. We compare our method with the aforementioned baselines on several tasks contained in Multi-Agent MuJoCo (see Table 7).

Fig. 8 demonstrates that, in all scenarios, DMS generally achieves superior performance over those of independent learning methods: IPPO and HAPPO-il. It is also worth noting that although HAPPO, one of the state-of-the-art centralized baselines, outperforms DMS, the performance gap between DMS and HAPPO is smaller than that of other IL algorithms in almost all tasks. Consequently, we can conclude that the DMS agents

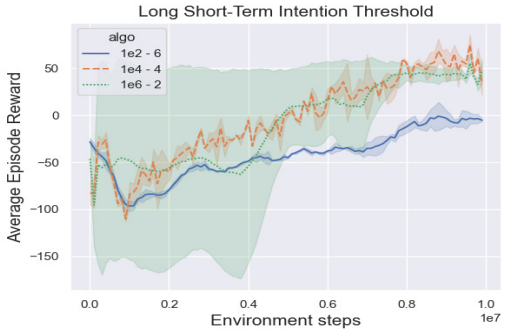


Figure 7: The effect of long short-term intention threshold on DMS's performance in Manyagent Swimmer scenario.

Table 7: Task configuration in Multi-Agent Mujoco.

Task	Agent Conf	Agent Obsk
2-Agent Ant	2x4	0
4-Agent Ant	4x2	0
8-Agent Ant	8x1	0
2-Agent HalfCheetah	2x3	0
3-Agent HalfCheetah	3x2	0
6-Agent HalfCheetah	6x1	0
2-Agent Walker	2x3	0
3-Agent Walker	3x2	0
6-Agent Walker	6x1	0
Manyagent Swimmer	10x2	0

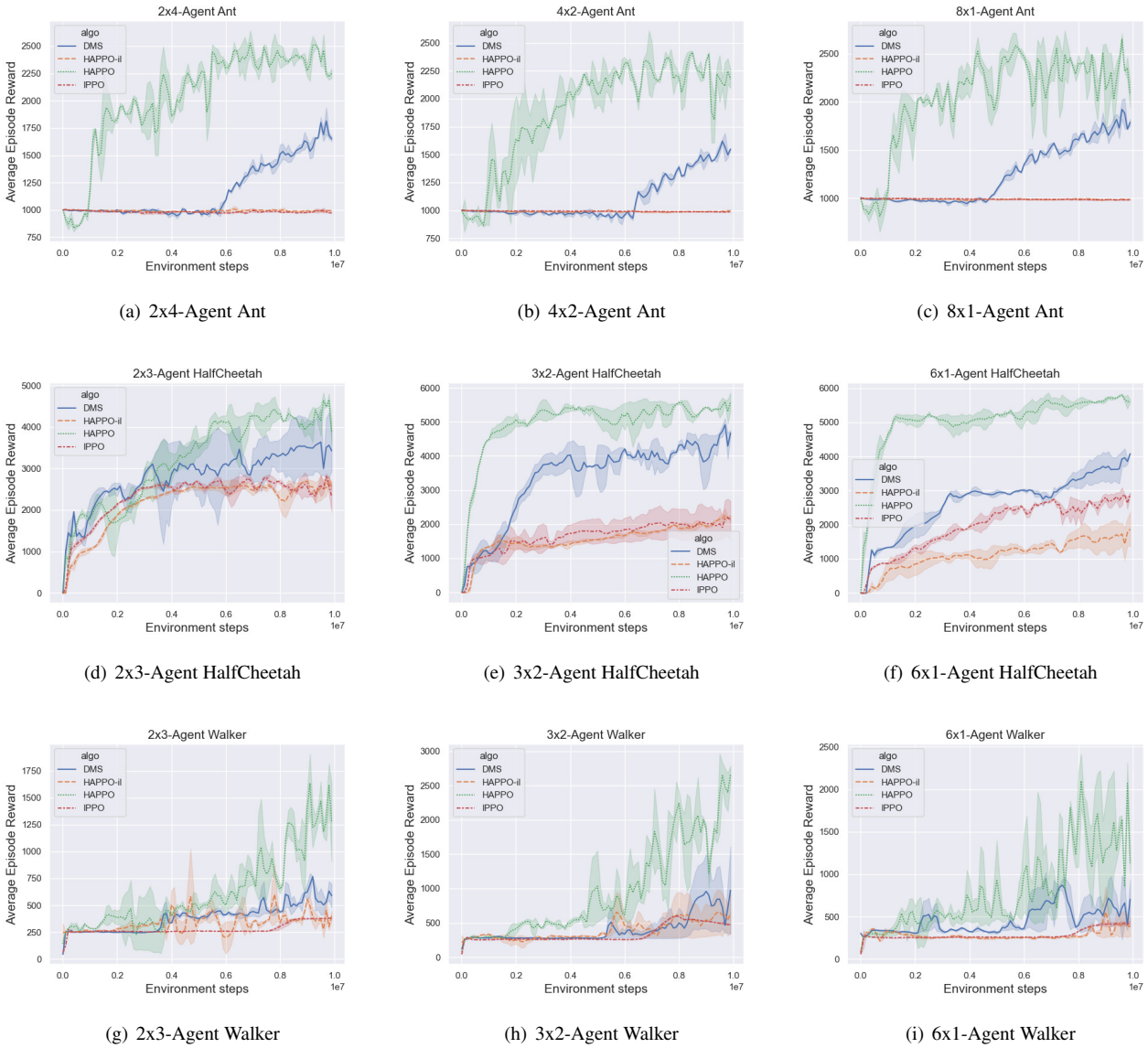
are able to learn coordination behavior in complex cooperative environments. The proposed method abstracts the correlation between agents and thus enhances the cooperation of agents.

5.4. Emerged Coordination Set

With adequate training from scratch, our agents would be able to learn coordinated behavior. In this section, we conduct a qualitative analysis on the correlation among agents. As shown in the heat map (see Fig. 9(a)) and screenshot (see Fig. 10(a)), in the initial stages of learning, the agents have a random correlation with all other agents and act in a rather uncoordinated way. Specifically, even if other agents enter within the field of view of the current agent, it will not be aware of the existence of others, let alone coordinate with others. As the training process continues, Fig.9(b) indicates that the correlation between the agents becomes stronger and more stable, and the coordination set emerges, as illustrated in Fig. 10(b).

5.5. Joint Observation and Individual Observation with Intention

In Section 4.1, we integrate the individual observations with the intentions of other agents to (partially) reconstruct the joint observation and mitigate partial observability. Here, we visualize the relationship between joint observation and individual observation with intention. As shown in Fig. 11, the enhanced individual observation of each agent and joint observation get spatially closer as training progresses, which means that each



(a) 2x4-Agent Ant (b) 4x2-Agent Ant (c) 8x1-Agent Ant

(d) 2x3-Agent HalfCheetah (e) 3x2-Agent HalfCheetah (f) 6x1-Agent HalfCheetah

(g) 2x3-Agent Walker (h) 3x2-Agent Walker (i) 6x1-Agent Walker

Figure 8: Performance comparison on multiple Multi-Agent MuJoCo tasks.

agent gradually learns the information about other agents' observation distributions.

5.6. Ablation Studies

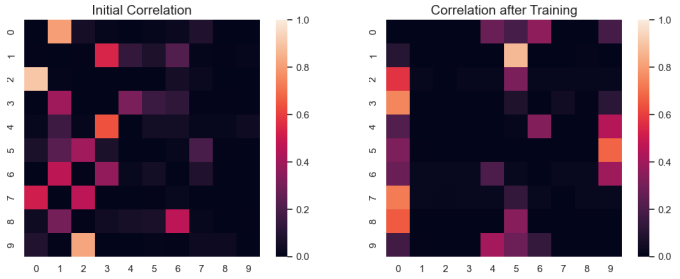
Our method is characterized by several contributions: Deep Motor System (DMS), agent-level coordination, and action masking. In this subsection, we design ablations to show their contributions.

5.6.1. Deep Motor System

We evaluate our model with and without the Deep Motor System (DMS) described in Section 4.1 and display the results in Fig. 12. DMS is one of the most critical parts of the model and has significant impacts on several SMAC maps and Multi-Agent MuJoCo tasks.

5.6.2. Agent-level Coordination

As mentioned in Section 4.2, we consider two coordination variants: (1) causal inference-based agent-level coordination (CI), in which the cooperative relationships are determined by inter-agent causal effects; (2) attention-based agent-level coordination (Attention), in which each agent coordinates with all other agents by a weighted average of each agent's intention, i.e., the coordination set of every agent contains all agents except itself. Fig. 13 illustrates that CI slightly outperforms Attention, which verifies that the causal effect among the agents captures high-order relationships and the necessity of coordination. Selective coordination forms more adaptable teams and accelerates the training process.



(a) Initial Correlation (b) Correlation after training

Figure 9: Heat map of correlation among agents. Each integer point on the x-axis and y-axis represents an agent. Each row shows the agent's relative correlations to other agents: the dark grid shows that two agents have a weak correlation, and the light grid shows that two agents have a strong correlation. The lighter the grid color is, the stronger correlation between this pair of agents. (The matrix is asymmetric as each agent independently judges the correlation to the other agents.)

(a) Initial Correlation (b) Correlation after training

Figure 10: Cooperation in StarCraftII

5.6.3. *Action Masking*

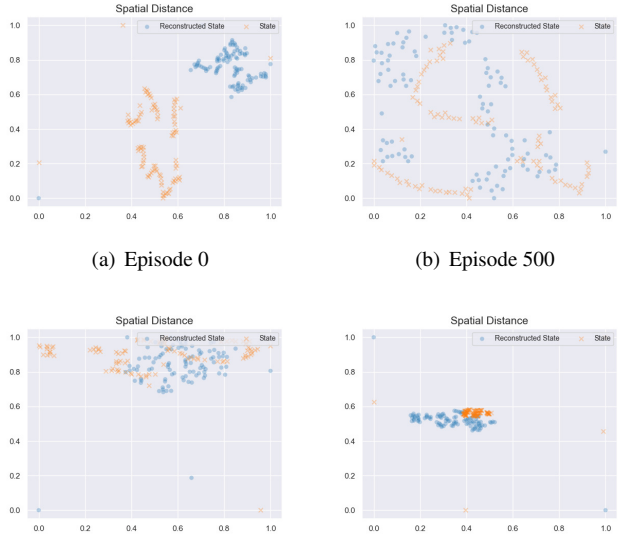
We finally analyze the justification of action masking. In the SMAC domain, as shown in Table 8, if the agent models the default action (that is, no operation), it will have a great impact on the intention modeling of others, because other agents prefer to be out of view (Fig. 1) rather than act nothing.

Table 8: The effect of action masking on DMS's win rate and standard deviation in SMAC.

Map	With Mask	Without Mask
3s5z	$97.5_{2.2}$	$63.3_{17.1}$
8m vs 9m	$82.6_{6.1}$	$61.6_{11.8}$
MMM2	$62.1_{8.8}$	$39.1_{34.3}$

6. Conclusion

We study the learning of coordinated behavior, which is a long-standing problem in cooperative MARL. We propose an inference-based independent learning MARL framework to solve this problem. Our work differs from the previous works focusing on the CTDE paradigm or communication. In order to alleviate non-stationarity, we first introduce Deep Motor System (DMS) to answer how to make agents aware of others. DMS allows agents to infer others' intentions only based on their individual motor systems without any other modifications. The question of how to choose partners to form a collaborative team is answered through the causal effects and attention between the agents, which accurately captures the necessity of cooperation. Empirically, we evaluate the proposed method on two multi-agent benchmarks (i.e., Multi-Agent MuJoCo and SMAC). Experimental results confirm that coordinated behavior among the agents can be learned in complex tasks even without the CTDE paradigm or communication. The results also demonstrate that our method makes agents learn consistent coordination policies during training and execution, which motivates us to extend this work to more realistic tasks. However, one limitation of our work is that we take more time per training epoch compared to the state-of-the-art MARL methods since we need additional training steps to update DMS. We believe that relaxing this limitation will be an important step for future work.



(a) Episode 0 (b) Episode 500

(c) Episode 1000 (d) Episode 1500

Figure 11: Spatial distance between joint observation and enhanced individual observation at training episode 0, 500, 1000, and 1500.

References

[1] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, T. Graepel, 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, AAMAS '18, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2018, p. 2085–2087.

[2] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, S. Whiteson, QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, Vol. 80

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

Figure 12: Effect of DMS.

(a) 2x3-Agent HalfCheetah (b) 3x2-Agent HalfCheetah (c) 6x1-Agent HalfCheetah

Figure 13: Effect of different coordination strategies.

of Proceedings of Machine Learning Research, PMLR, 2018, pp. 4295–4304.
URL https://proceedings.mlr.press/v80/rashid18a.html

[3] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, Y. Yi, QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, Vol. 97 of Proceedings of Machine Learning Research, PMLR, 2019, pp. 5887–5896.
URL https://proceedings.mlr.press/v97/son19a.html

[4] Z. Ding, T. Huang, Z. Lu, Learning individually inferred communication for multi-agent cooperation, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 22069–22079.
URL https://proceedings.neurips.cc/paper/2020/file/fb2fcd534b0ff3bbed73cc51df620323-Paper.pdf

[5] S. Q. Zhang, Q. Zhang, J. Lin, Succinct and robust multi-agent communication with temporal message control, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 17271–17282.
URL https://proceedings.neurips.cc/paper/2020/file/c82b013313066e0702d58dc70db033ca-Paper.pdf

[6] W. Boehmer, V. Kurin, S. Whiteson, Deep coordination graphs, in: H. D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, Vol. 119 of Proceedings of Machine Learning Research, PMLR, 2020, pp. 980–991.
URL https://proceedings.mlr.press/v119/boehmer20a.html

[7] T. Wang, L. Zeng, W. Dong, Q. Yang, Y. Yu, C. Zhang, Context-aware sparse deep coordination graphs (2021). doi:10.48550/ARXIV.2106.02886.
URL https://arxiv.org/abs/2106.02886

[8] V. Müller, K.-R. P. Ohström, U. Lindenberger, Interactive brains, social minds: Neural and physiological mechanisms of interpersonal action coordination, Neuroscience Biobehavioral Reviews 128 (2021) 661–677. doi:https://doi.org/10.1016/j.neubiorev.2021.07.017.
URL https://www.sciencedirect.com/science/article/pii/S014976342100316X

[9] D. Y.-J. Yang, G. Rosenblau, C. Keifer, K. A. Pelphrey, An integrative neural model of social perception, action observation, and theory of mind, Neuroscience Biobehavioral Reviews 51 (2015) 263–275. doi:https://doi.org/10.1016/j.neubiorev.2015.01.020.
URL https://www.sciencedirect.com/science/article/pii/S0149763415000317

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017.
URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations, 2021.
URL https://openreview.net/forum?id=YicbFdNTTy

[12] C. S. de Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. S. Torr, M. Sun, S. Whiteson, Is independent learning all you need in the starcraft multi-agent challenge? (2020). doi:10.48550/ARXIV.2011.09533.
URL https://arxiv.org/abs/2011.09533

[13] J. G. Kuba, R. Chen, M. Wen, Y. Wen, F. Sun, J. Wang, Y. Yang, Trust region policy optimisation in multi-agent reinforcement learning (2021). doi:10.48550/ARXIV.2109.11251.
URL https://arxiv.org/abs/2109.11251

[14] M. Tan, Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997, p. 487–494.

[15] Y. Wang, B. Han, T. Wang, H. Dong, C. Zhang, Off-policy multi-agent

decomposed policy gradients (2020). doi:10.48550/ARXIV.2007.12322.
URL https://arxiv.org/abs/2007.12322

[16] P. Wang, J. Wang, P. Paranamana, P. Shafto, A mathematical theory of cooperative communication, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20, Curran Associates Inc., Red Hook, NY, USA, 2020.

[17] H. Zhang, W. Chen, Z. Huang, M. Li, Y. Yang, W. Zhang, J. Wang, Bi-level actor-critic for multi-agent coordination, Proceedings of the AAAI Conference on Artificial Intelligence 34 (05) (2020) 7325–7332. doi:10.1609/aaai.v34i05.6226.
URL https://ojs.aaai.org/index.php/AAAI/article/view/6226

[18] W. Kim, J. Park, Y. Sung, Communication in multi-agent reinforcement learning: Intention sharing, in: International Conference on Learning Representations, 2021.
URL https://openreview.net/forum?id=qpsl2dR9twy

[19] T. Lin, J. Huh, C. Stauffer, S. N. Lim, P. Isola, Learning to ground multi-agent communication with autoencoders, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems, Vol. 34, Curran Associates, Inc., 2021, pp. 15230–15242.
URL https://proceedings.neurips.cc/paper/2021/file/80fee67c8a4c4989bf8a580b4bbb0cd2-Paper.pdf

[20] M. Rangwala, R. Williams, Learning multi-agent communication through structured attentive reasoning, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 10088–10098.
URL https://proceedings.neurips.cc/paper/2020/file/72ab54f9b8c11fae5b923d7f854ef06a-Paper.pdf

[21] R. Wang, X. He, R. Yu, W. Qiu, B. An, Z. Rabinovich, Learning efficient multi-agent communication: An information bottleneck approach, in: H. D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, Vol. 119 of Proceedings of Machine Learning Research, PMLR, 2020, pp. 9908–9918.
URL https://proceedings.mlr.press/v119/wang20i.html

[22] J. P. Inala, Y. Yang, J. Paulos, Y. Pu, O. Bastani, V. Kumar, M. Rinard, A. Solar-Lezama, Neurosymbolic transformers for multi-agent communication, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 13597–13608.
URL https://proceedings.neurips.cc/paper/2020/file/9d740bd0f36aaa312c8d504e28c42163-Paper.pdf

[23] J. Ruan, Y. Du, X. Xiong, D. Xing, X. Li, L. Meng, H. Zhang, J. Wang, B. Xu, Gcs: Graph-based coordination strategy for multi-agent reinforcement learning (2022). doi:10.48550/ARXIV.2201.06257.
URL https://arxiv.org/abs/2201.06257

[24] N. Jaques, A. Lazaridou, E. Hughes, C. Gulcehre, P. Ortega, D. Strouse, J. Z. Leibo, N. De Freitas, Social influence as intrinsic motivation for multi-agent deep reinforcement learning, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, Vol. 97 of Proceedings of Machine Learning Research, PMLR, 2019, pp. 3040–3049.
URL https://proceedings.mlr.press/v97/jaques19a.html

[25] C. Schroeder de Witt, J. Foerster, G. Farquhar, P. Torr, W. Boehmer, S. Whiteson, Multi-agent common knowledge reinforcement learning, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019.
URL https://proceedings.neurips.cc/paper/2019/file/f968fdc88852a4a3a27a81fe3f57bfc5-Paper.pdf

[26] Y. Zhang, Q. Yang, D. An, C. Zhang, Coordination between individual agents in multi-agent reinforcement learning, Proceedings of the AAAI Conference on Artificial Intelligence 35 (13) (2021) 11387–11394.
URL https://ojs.aaai.org/index.php/AAAI/article/view/17357

[27] D. Strouse, M. Kleiman-Weiner, J. Tenenbaum, M. Botvinick, D. J. Schwab, Learning to share and hide intentions using information regularization, in: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 31, Curran Associates, Inc., 2018.
URL https://proceedings.neurips.cc/paper/2018/file/1ef03ed0cd5863c550128836b28ec3e9-Paper.pdf

[28] Z. Tian, S. Zou, I. Davies, T. Warr, L. Wu, H. B. Ammar, J. Wang, Learning to communicate implicitly by actions, Proceedings of the AAAI Conference on Artificial Intelligence 34 (05) (2020) 7261–7268. doi:10.1609/aaai.v34i05.6217.
URL https://ojs.aaai.org/index.php/AAAI/article/view/6217

[29] J. G. Kuba, M. Wen, L. Meng, s. gu, H. Zhang, D. Mguni, J. Wang, Y. Yang, Settling the variance of multi-agent policy gradients, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems, Vol. 34, Curran Associates, Inc., 2021, pp. 13458–13470.
URL https://proceedings.neurips.cc/paper/2021/file/6fe6a8a6e6cb710584efc4af0c34ce50-Paper.pdf

[30] J. Schulman, P. Moritz, S. Levine, M. Jordan, P. Abbeel, High-dimensional continuous control using generalized advantage estimation (2015). doi:10.48550/ARXIV.1506.02438.
URL https://arxiv.org/abs/1506.02438

[31] G. Rizzolatti, L. Cattaneo, M. Fabbri-Destro, S. Rozzi, Cortical mechanisms underlying the organization of goal-directed actions and mirror neuron-based action understanding, Physiological Reviews 94 (2) (2014) 655–706, pMID: 24692357. arXiv:https://doi.org/10.1152/physrev.00009.2013, doi:10.1152/physrev.00009.2013.
URL https://doi.org/10.1152/physrev.00009.2013

[32] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization (2014). doi:10.48550/ARXIV.1412.6980.
URL https://arxiv.org/abs/1412.6980

[33] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078 (2014).

[34] J. L. Ba, J. R. Kiros, G. E. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450 (2016).

[35] D. P. Kingma, T. Salimans, M. Welling, Variational dropout and the local reparameterization trick, in: C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 28, Curran Associates, Inc., 2015.
URL https://proceedings.neurips.cc/paper/2015/file/bc7316929fe1545bf0b98d114ee3ecb8-Paper.pdf

[36] L. Zettlemoyer, B. Milch, L. Kaelbling, Multi-agent filtering with infinitely nested beliefs, in: D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (Eds.), Advances in Neural Information Processing Systems, Vol. 21, Curran Associates, Inc., 2008.
URL https://proceedings.neurips.cc/paper/2008/file/6c3cf77d52820cd0fe646d38bc2145ca-Paper.pdf

[37] M. Samvelyan, T. Rashid, C. S. de Witt, G. Farquhar, N. Nardelli, T. G. J. Rudner, C.-M. Hung, P. H. S. Torr, J. Foerster, S. Whiteson, The starcraft multi-agent challenge (2019). doi:10.48550/ARXIV.1902.04043.
URL https://arxiv.org/abs/1902.04043

[38] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning (2015). doi:10.48550/ARXIV.1509.02971.
URL https://arxiv.org/abs/1509.02971

[39] S. Fujimoto, H. van Hoof, D. Meger, Addressing function approximation error in actor-critic methods, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, Vol. 80 of Proceedings of Machine Learning Research, PMLR, 2018, pp. 1587–1596.
URL https://proceedings.mlr.press/v80/fujimoto18a.html

[40] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, Vol. 27, Curran Associates, Inc., 2014.
URL https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[41] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms (2017). doi:10.48550/ARXIV.1707.06347.
URL https://arxiv.org/abs/1707.06347