

Full Length Article

An off-policy multi-agent stochastic policy gradient algorithm for cooperative continuous control[☆]

Delin Guo^a, Lan Tang^{a,*}, Xinggan Zhang^a, Ying-chang Liang^b

^a School of Electronic Science and Engineering, Nanjing University, Nanjing, 210093, China

^b Center for Intelligent Networking and Communications, University of Electronic Science and Technology of China, Chengdu, 611731, China

ARTICLE INFO

Keywords:

Multi-agent control
Deep reinforcement learning (DRL)
Multi-agent reinforcement learning (MARL)
Trust region
Multi-agent MuJoCo

ABSTRACT

Multi-agent reinforcement learning (MARL) algorithms based on trust regions (TR) have achieved significant success in numerous cooperative multi-agent tasks. These algorithms restrain the Kullback–Leibler (KL) divergence (i.e., TR constraint) between the current and new policies to avoid aggressive update steps and improve learning performance. However, the majority of existing TR-based MARL algorithms are on-policy, meaning that they require new data sampled by current policies for training and cannot utilize off-policy (or historical) data, leading to low sample efficiency. This study aims to enhance the data efficiency of TR-based learning methods. To achieve this, an approximation of the original objective function is designed. In addition, it is proven that as long as the update size of the policy (measured by the KL divergence) is restricted, optimizing the designed objective function using historical data can guarantee the monotonic improvement of the original target. Building on the designed objective, a practical off-policy multi-agent stochastic policy gradient algorithm is proposed within the framework of centralized training with decentralized execution (CTDE). Additionally, policy entropy is integrated into the reward to promote exploration, and consequently, improve stability. Comprehensive experiments are conducted on a representative benchmark for multi-agent MuJoCo (MAMuJoCo), which offers a range of challenging tasks in cooperative continuous multi-agent control. The results demonstrate that the proposed algorithm outperforms all other existing algorithms by a significant margin.

1. Introduction

1.1. Backgrounds

The combination of deep neural networks (DNNs) and conventional reinforcement learning (RL) methods has led to the emergence of deep reinforcement learning (DRL) in recent years. Because of the strong representational capacity of DNNs, DRL has demonstrated remarkable success in various domains. For example, Silver et al. (2016) combined the Monte Carlo Tree Search (MCTS) with DNN to attain master-level performance in the game of Go. Deep Q Network (DQN) (Mnih et al., 2015) has attained human-level control in video games. Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2016) demonstrated strong performance in the continuous control of robotics. Shim and Li (2017) introduced a spiking neural network (SNN)-based Q-learning algorithm that offers a more biologically plausible approach and achieves superior performance compared with Q-learning in collision avoidance tasks.

The previously introduced methods were designed for tasks involving only one agent, and thus are referred to as single-agent RL algorithms. Real-world scenarios often involve multiple interacting agents. This study is dedicated to the design of an algorithm tailored for cooperative multi-agent tasks such as networking packet routing problems (Ye, Zhang, & Yang, 2015), traffic light coordination (Van der Pol & Oliehoek, 2016), robot swarm control (Hüttenrauch, Šošić, & Neumann, 2017), and coordination tasks involving unmanned aerial vehicles (UAVs) (Cao, Yu, Ren, & Chen, 2013). By treating all agents as a single entity, single-agent RL methods offer a straightforward approach for training a centralized policy responsible for decision-making on behalf of all agents. Nevertheless, practical situations may have constraints such as partial visibility, communication limitations, and privacy concerns, which can render fully centralized control impractical. In addition, even with full observation, training a set of decomposed policies with a small action space is often easier than training a centralized policy with a high-dimensional action space for practical implementation (Weisz, Budzianowski, Su, & Gašić, 2018).

[☆] This work was supported by the National Natural Science Foundation of China under grants 62072229, U1936201, 62071220, 61976113.

* Corresponding author.

E-mail addresses: guodelin@smail.nju.edu.cn (D. Guo), tanglan@nju.edu.cn (L. Tang), zhxg@nju.edu.cn (X. Zhang), liangyc@ieee.org (Y.-c. Liang).

Consequently, these factors necessitate the training of decentralized policies that enable agents to operate in a distributed manner.

The most straightforward way to acquire decentralized policies is to train each agent independently and consider other agents as part of the environment, as demonstrated by independent Q-learning (Tan, 1993). However, this paradigm often results in an unstable training process. This is because, from the viewpoint of any agent, the environment becomes nonstationary because of the changing policies of other agents, thereby violating the stationary assumption necessary for RL algorithms to converge. To address this challenge, multi-agent DDPG (MADDPG) (Lowe et al., 2017) and counterfactual multi-agent policy gradient (COMA) (Foerster, Farquhar, Afouras, Nardelli, & Whiteson, 2018) employed a centralized value function (trained on global information) to guide the training of decentralized policies. This framework, known as centralized training with decentralized execution (CTDE), has gained popularity and become a standard framework in cooperative MARL, inspiring a series of related studies (Chen, Li, Liu, Yang, & Gao, 2023; Guo, Tang, Zhang, & Liang, 2020; Hu, Hu, & Liao, 2022; Iqbal & Sha, 2019; Kuba et al., 2022; Li & He, 2020; Peng et al., 2021; Rashid, Farquhar, Peng, & Whiteson, 2020; Rashid et al., 2018; Son, Kim, Kang, Hostallero, & Yi, 2019; Su, Adams, & Beling, 2021; Sunehag et al., 2018; Wang et al., 2021; Wang, Han, Wang, Dong, & Zhang, 2020; Wang, Ren, Liu, Yu, & Zhang, 2020; Wang, Zhang, Kim, & Gu, 2020; Wen et al., 2022; Yang et al., 2020; Yu et al., 2022; Zhao, Zhu, Liu, Wang, & Chen, 2023; Zhong, Kuba, Hu, Ji, & Yang, 2023; Zhou, Liu, Sui, Li, & Chung, 2020).

MARL algorithms can be categorized into two primary classes: value-based and policy-based. Most existing value-based MARL algorithms follow a common methodology, which involves decomposing the joint action-value function into individual action-value functions. The most representative examples of this approach are Value Decomposition Network (VDN) (Sunehag et al., 2018) and QMIX (Rashid et al., 2018). VDN simply decomposes the joint action-value function into the sum of individual action-value functions, whereas QMIX represents the joint action-value function as a monotonic mixing function of individual action-value functions, offering broader applicability to multi-agent tasks compared with VDN. Following these two pioneering studies, numerous value-based MARL algorithms were developed using the value decomposition (VD) technique (Rashid et al., 2020; Son et al., 2019; Wang et al., 2021; Wang, Ren, et al., 2020; Yang et al., 2020; Zhao et al., 2023). Despite their achievements, VD-based methods have certain noteworthy limitations. The VD-based methods are based on the Individual-Global-Max (IGM) assumption (Son et al., 2019). That is, the global greedy action is equivalent to the ensemble of individual greedy actions, which is a very strong assumption that is not always satisfied in some tasks. As reported in Hu, Jiang, Harding, Wu, and Liao (2021), QMIX fails in a simple cooperative matrix game when IGM properties are not satisfied. Second, value-based MARL algorithms can only address problems with low-dimensional (often one-dimensional) discrete action spaces. Finally, the degree of centralization in value-based MARL algorithms is typically higher than that in policy-based methods¹. Although these studies considered the CTDE framework, the prevailing preference remains to share as little information as possible among agents.

The policy-based methods also adhere to a similar paradigm, employing a centralized critic to guide the update of decentralized policies (e.g., Foerster et al., 2018; Lowe et al., 2017). In Iqbal and Sha (2019), the multi-attention actor-critic (MAAC) was proposed, which integrates the attention mechanism into the centralized critic to enhance scalability and effectiveness in complex multi-agent tasks. Su et al. (2021) introduced the value decomposition actor-critic (VDAC), where the

centralized critic is represented by mixing factored value functions (similar to QMIX). Extending MADDPG, Peng et al. (2021) represents the centralized critic similarly to VDAC but removes the monotonic constraint from the mixer, claiming enhanced representational capacity. To address issues of centralized–decentralized mismatch, Wang, Han, et al. (2020) presented a decomposed policy gradient algorithm (DOP) adopting the VD technique. Investigating credit assignment problem in MARL, Wang, Zhang, et al. (2020) proposed Shapley Q-value DDPG (SQDDPG), allocating centralized Q-value to each agent based on Shapley value method. Additionally, Zhou et al. (2020) formulates Learning Implicit Credit Assignment (LICA) as a hyper-network for representing the centralized critic and integrating implicit information of the global state into policy gradients through multiplicative association with decentralized policies.

Policy-based methods adhere to a similar paradigm by employing a centralized critic to guide the update of decentralized policies (e.g., Foerster et al., 2018; Lowe et al., 2017). In Iqbal and Sha (2019), the multi-attention actor-critic (MAAC) was proposed, which integrates the attention mechanism into the centralized critic to enhance the scalability and effectiveness in complex multi-agent tasks. Su et al. (2021) introduced the actor-critic value decomposition (VDAC), in which the centralized critic is represented by mixing factored value functions (similar to QMIX). Extending MADDPG, Peng et al. (2021) represents a centralized critic similar to VDAC but removes the monotonic constraint from the mixer, claiming enhanced representational capacity. To address the issues of centralized–decentralized mismatch, Wang, Han, et al. (2020) presented a decomposed policy gradient algorithm (DOP) adopting the VD technique. Investigating the credit assignment problem in MARL, Wang, Zhang, et al. (2020) proposed Shapley Q-value DDPG (SQDDPG), which allocates a centralized Q-value to each agent based on Shapley value method. Additionally, Zhou et al. (2020) formulated the Learning Implicit Credit Assignment (LICA) as a hypernetwork to represent the centralized critic and integrate implicit information of the global state into policy gradients through multiplicative association with decentralized policies.

The aforementioned policy-based methods can address issues related to multidimensional continuous action spaces and require less information sharing during centralized training. Moreover, these methods are often theoretically justified, as demonstrated by the proven convergence properties of Foerster et al. (2018), Su et al. (2021) and Wang, Han, et al. (2020). Based on these advantages, policy-based MARL algorithms are expected to outperform and be more applicable than value-based approaches. However, as reported in Hu et al. (2021), many of these methods (Peng et al., 2021; Su et al., 2021; Wang, Han, et al., 2020; Zhou et al., 2020) have exhibited poor performance. This could be attributed to occasional aggressive policy updates. The convergence condition of the actor-critic framework suggests that the update timescales for the critic and actor should be sufficiently slow; further, the actor should be updated sufficiently slower than the critic (Foerster et al., 2018). However, the second half of the condition is overly strict and can significantly impede the learning process. Consequently, practical algorithms typically adopt a moderate learning rate scale (e.g., $[10^{-4}, 10^{-3}]$). Nevertheless, even with a moderate update size, instances of aggressive updates may still exist, leading to poor policy updates and instability. Furthermore, within the actor-critic framework, where both policy and value functions are trained jointly and influence each other's updates, incorrect policy updates can misguide value function updates, and misguided value functions can subsequently mislead further policy updates. Therefore, the instability caused by erroneous policy updates will be further amplified within this coupled learning framework. Previous studies on policy-based methods have primarily focused on designing critics. However, the stability of policy updates has been neglected, which may critically contribute to their underperformance.

The above discussion suggests that ensuring stable policy updates may be a key factor in improving the performance of policy-based

¹ For example, the agents in the policy-based COMA (Foerster et al., 2018) only need to share their actions in training. While in QMIX, the parameters and gradients of the individual action-value functions are also shared.

algorithms. However, simply using smaller learning rates for policy updates is insufficient because they cannot guard against occasional aggressive updates while maintaining the desired learning speed (Wang et al., 2017). To address the dilemma between learning stability and speed, Schulman, Levine, Abbeel, Jordan, and Moritz (2015) proposed single-agent trust region policy optimization (TRPO), providing a safe and robust way to update policy. Specifically, Schulman et al. (2015) constrained the Kullback–Leibler (KL) divergence (i.e., the trust region constraint) between the current and new policies to avoid aggressive updates to stabilize the learning process. In addition, a monotonic improvement guarantee for TRPO is provided. Following TRPO, a series of TR-based successors (e.g., Meng, Zheng, Shi, & Pan, 2022; Schulman, Wolski, Dhariwal, Radford, & Klimov, 2017; Wang et al., 2017; Wang, He, & Tan, 2020; Wu, Mansimov, Grosse, Liao, & Ba, 2017) have been developed and have achieved superior performance on a range of single-agent tasks, the most well-known of which is probably proximal policy optimization (PPO) in Schulman et al. (2017).

Inspired by the success of trust region (TR)-based methods in single-agent domains, Guo et al. (2020) extended PPO to multi-agent PPO (MAPPO) by introducing the CTDE framework, achieving good performance in a handover control problem in a wireless communication network. Concurrently, Yu et al. (2022) conducted extensive experiments on MAPPO and demonstrated its superiority over other approaches for multi-agent StarCraft tasks (Samvelyan et al., 2019). Building upon this work, Hu et al. (2022) further improved MAPPO with minor modifications to the critic input. Additionally, Li and He (2020) proposed a Multi-Agent TRPO (MATRPO), which exhibits a learning performance comparable to that of a fully centralized TRPO. Heterogeneous-agent TRPO (HATRPO) and Heterogeneous-Agent PPO (HAPPO) were introduced in Kuba et al. (2022), where HATRPO was proven to have a monotonic improvement guarantee and claimed to achieve state-of-the-art performance on multi-agent MuJoCo (MaMuJoCo) benchmarks (Peng et al., 2021), which are known for their challenging nature in continuous control settings. Multi-agent trust region optimization (MATRO), proposed in Wen et al. (2022), leverages Nash equilibrium concepts and demonstrates strong performance across MaMuJoCo environments, switch grid worlds, and Atari games. The heterogeneous-agent reinforcement learning (HARL) framework was presented in Zhong et al. (2023), which strengthened the theoretical explanations for HATRPO and HAPPO.

1.2. Motivation and contributions

By incorporating the TR constraint, policy-based MARL methods have demonstrated improved learning performance. In conjunction with the advantages outlined in the previous subsection, TR-based MARL algorithms possess greater power than their value-based counterparts. However, all the existing TR-based MARL algorithms are operated in an on-policy fashion. This implies that each gradient update requires newly sampled data from current policies, which in turn limits the sample efficiency. In addition, these TR-based methods may suffer from poor exploration, leading to a higher likelihood of becoming stuck in local optima and also makes them brittleness concerning hyperparameters.

Motivated by the above challenges, this study aims to improve the sampling efficiency and exploration capability of TR-based methods. An entropy-augmented objective is formulated to enhance policy exploration. Because of the intractability of the original objective, a novel surrogate objective function that approximates the original objective is proposed. The main advantages of the proposed objective are threefold. (1) The joint original objective is approximated as the sum of the decomposed individual objectives of each agent without requiring the IGM assumption. (2) Off-policy data can be used to achieve the proposed objectives. (3) A critical theorem proves that optimizing the surrogate objective under the KL divergence constraint guarantees a monotonic improvement of the original objective. Based

on the proposed surrogate objective, a practical off-policy MARL algorithm is presented. The reparameterization technique is adopted to obtain a low-variance estimator. The update size of the policy was controlled by introducing a KL penalty term, and the penalty factor is automatically adjusted to achieve the desired KL divergence value. Finally, comprehensive experiments on the MaMuJoCo benchmark are conducted to demonstrate the superiority of the proposed method. The simulation results highlight that the approach significantly outperforms all other algorithms in terms of average return and sample efficiency. The main contributions of this study are summarized as follows.

- A novel surrogate objective function is developed, that can utilize off-policy data in training. The scaled policy entropy is added to the reward function to enhance policy exploration. It is proved that the monotonic improvement of this surrogate objective function can guarantee the monotonic improvement of policies.
- Based on the proposed objective, a practical MARL algorithm with off-policy implementation is proposed. Comprehensive experiments on the benchmark MaMuJoCo are conducted to show the superiority of the proposed method.

The remainder of this paper is organized as follows. Some necessary backgrounds and definitions are introduced in Section 2. Section 3 presents the proposed objective and relevant analyses. Section 4 presents the practical implementation of the proposed algorithm. Numerical simulations and a discussion of the developed algorithm are presented in Section 5. Finally, Section 6 concludes this study and introduces directions for our future work.

2. Preliminaries

In this section, the relevant background information on MARL is provided.

2.1. Cooperative Markov game

A fully cooperative Markov game (Littman, 1994) with N agents is considered, which is described by a tuple $\langle N, S, \mathcal{A}, P, r, \gamma \rangle$. The agent index set is denoted by $N = \{1, 2, \dots, N\}$, S represents the system state space, and $\mathcal{A} = \prod_{i=1}^N \mathcal{A}^i$ is the Cartesian product of the action spaces of all agents, also known as the joint action space. $P : S \times \mathcal{A} \rightarrow \mathcal{P}(S)$ denotes the transition probability function, where $\mathcal{P}(S)$ represents the set of probability measures on S . $r : S \times \mathcal{A} \rightarrow \mathbb{R}$ denotes the shared reward function. Additionally, $\gamma \in [0, 1)$ denotes the discount factor.

In particular, at each time step t , the system state is denoted by $s_t \in S$. Each agent i executes action $a_t^i \in \mathcal{A}^i$ according to its individual policy $\pi^i(\cdot|s_t)$. The collection of individual actions forms a joint action $\mathbf{a}_t = (a_t^1, \dots, a_t^N)$. Subsequently, the agents receive the shared reward $r_t = r(s_t, \mathbf{a}_t)$ and the system transitions to the next state $s_{t+1} \sim P(\cdot|s_t, \mathbf{a}_t)$.

The standard action-value, state-value, and the advantage functions are defined as follows:

$$Q^\pi(s, \mathbf{a}) = \mathbb{E}_{\tau \sim \pi} \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, \mathbf{a}_t = \mathbf{a} \right\} \quad (1a)$$

$$V^\pi(s) = \mathbb{E}_{\tau \sim \pi} \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s \right\} \quad (1b)$$

$$A^\pi(s, \mathbf{a}) = Q^\pi(s, \mathbf{a}) - V^\pi(s), \quad (1c)$$

respectively, where $\pi = \prod_{i=1}^N \pi^i$ is the joint policy of all agents and $\tau = (s_t, \mathbf{a}_t, s_{t+1}, \dots)$ indicates the trajectory generated by applying π . In (MA)RL, the target is to maximize the expected value function

$$\max_{\pi^1, \dots, \pi^N} \eta(\pi) = \mathbb{E}_{s_0 \sim p_0} \{V^\pi(s_0)\}, \quad (2)$$

where $p_0(s)$ is the initial state distribution.

For an arbitrary policy $\tilde{\pi}$ and π , the following holds (Schulman et al., 2015)

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s \sim p_\pi, \mathbf{a} \sim \tilde{\pi}} \{A^\pi(s, \mathbf{a})\}, \quad (3)$$

where $\rho_{\pi}(s) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s | \pi)$ is the (unnormalized) discounted state distribution.

Partially observable: In some scenarios, the global state s_t may be unavailable. For example, in poker games, even teammates' cards may be unknown. In such cases, each agent $i \in \mathcal{N}$ can only make decisions based on its local observation o'_t , that is, $a'_t \sim \pi^i(\cdot | o'_t)$. In this study, the analysis is conducted within a fully observable setting, whereas partially observable scenarios are discussed in the experimental section.

2.2. CTDE

Centralized Training with Decentralized Execution (CTDE) is widely regarded as the predominant paradigm in cooperative multi-agent tasks. Its application in MARL was pioneered in [Lowe et al. \(2017\)](#) and [Foerster et al. \(2018\)](#). CTDE enables agents to train their individual policies using global information, including actions taken by other agents, and allows the sharing of policy parameters and gradients. However, during the execution phase, agents are constrained to make decisions based solely on their decentralized policies without any provision for information sharing. This study aims to develop an MARL algorithm that operates within the CTDE framework.

2.3. TRPO and off-policy TRPO

In this subsection, two single-agent RL algorithms, TRPO ([Schulman et al., 2015](#)) and off-policy TRPO ([Meng et al., 2022](#)), are briefly introduced.

It can be observed from Eq. (3) that if $\mathbb{E}_{s \sim \rho_{\pi}, a \sim \pi} \{A^{\pi}(s, a)\} \geq 0$, updating from π to $\tilde{\pi}$ will lead to an improvement of $\eta(\pi)$. However, because of the implicit form of $\rho_{\pi}(s)$ and also the intricate dependency of $\rho_{\pi}(s)$ on π , optimizing Eq. (3) directly is impractical. Therefore, an approximation to $\eta(\tilde{\pi})$ is proposed by replacing $\rho_{\pi}(s)$ with $\rho_{\pi}(s)$ as follows:

$$L_{\pi}(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s \sim \rho_{\pi}, a \sim \tilde{\pi}} \{A^{\pi}(s, a)\}. \quad (4)$$

Let $D_{\text{KL}}(\cdot \| \cdot)$ denote the Kullback–Leibler (KL) divergence, which quantifies the similarity between two probability distributions. More clearly, $D_{\text{KL}}(\pi(\cdot | s) \| \tilde{\pi}(\cdot | s)) = \sum_{a \in \mathcal{A}} \pi(a | s) \log \frac{\pi(a | s)}{\tilde{\pi}(a | s)}$. Besides, define $D_{\text{KL}}^{\max}(\pi, \tilde{\pi}) = \max_s D_{\text{KL}}(\pi(\cdot | s) \| \tilde{\pi}(\cdot | s))$. The relationship between $L_{\pi}(\tilde{\pi})$ and $\eta(\tilde{\pi})$ is given by [Schulman et al. \(2015\)](#)

$$\eta(\tilde{\pi}) \geq L_{\pi}(\tilde{\pi}) - \frac{4\epsilon\gamma}{(1-\gamma)^2} D_{\text{KL}}^{\max}(\pi, \tilde{\pi}), \quad (5)$$

where $\epsilon = \max_{s,a} |A^{\pi}(s, a)|$, Eq. (5) suggests that maximizing the right-hand side improves the original objective.

In order to achieve larger and more robust update sizes, [Schulman et al. \(2015\)](#) relaxed the problem, $\max_{\tilde{\pi}} L_{\pi}(\tilde{\pi}) - \frac{4\epsilon\gamma}{(1-\gamma)^2} D_{\text{KL}}^{\max}(\pi, \tilde{\pi})$, by transforming the penalty associated with the maximum KL divergence into a constraint involving the average KL. The relaxed problem is given by

$$\max_{\tilde{\pi}} L_{\pi}(\tilde{\pi}) \quad (6a)$$

$$\text{s.t.} : \bar{D}_{\text{KL}}^{\rho_{\pi}}(\pi, \tilde{\pi}) \leq \delta, \quad (6b)$$

where $\bar{D}_{\text{KL}}^{\rho_{\pi}}(\pi, \tilde{\pi}) = \mathbb{E}_{s \sim \rho_{\pi}} \{D_{\text{KL}}(\pi(\cdot | s) \| \tilde{\pi}(\cdot | s))\}$ is the average KL divergence. Finally, [Schulman et al. \(2015\)](#) provided an approximate solution to the problem (6) by employing a linear approximation of the objective function and a quadratic approximation of the KL constraint. Using this approximation, a closed-form formula for policy updating was derived.

To enhance sample efficiency, the off-policy TRPO in [Meng et al. \(2022\)](#) further proposed the following local approximation to $\eta(\tilde{\pi})$:

$$L_{\pi,b}(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s \sim \rho_b, a \sim \tilde{\pi}} \{A^{\pi}(s, a)\}, \quad (7)$$

where ρ_b is the discounted state distribution generated by an arbitrary behavior policy b . Eq. (7) indicates that off-policy data can be used

to estimate $L_{\pi,b}(\tilde{\pi})$. The following relationship is given in [Meng et al. \(2022\)](#):

$$\eta(\tilde{\pi}) \geq L_{\pi,b}(\tilde{\pi}) - \frac{4\epsilon\gamma}{(1-\gamma)^2} \left(D_{\text{KL}}^{\max}(\pi, \tilde{\pi}) + \sqrt{D_{\text{KL}}^{\max}(b, \pi) D_{\text{KL}}^{\max}(\pi, \tilde{\pi})} \right), \quad (8)$$

Finally, similar to the on-policy TRPO, the policy is updated by repeatedly solving the following optimization problem

$$\max_{\tilde{\pi}} L_{\pi,b}(\tilde{\pi}) \quad (9a)$$

$$\text{s.t.} : \sqrt{\bar{D}_{\text{KL}}^{\rho_b}(\pi, \tilde{\pi}) \cdot \bar{D}_{\text{KL}}^{\rho_b}(b, \pi) + \bar{D}_{\text{KL}}^{\rho_b}(\pi, \tilde{\pi})} \leq \delta, \quad (9b)$$

The solution to (9) is similar to that of the on-policy TRPO in (6).

3. Off-policy multi-agent policy gradient

To enhance the stability and exploration, as in [Haarnoja, Zhou, Abbeel, and Levine \(2018\)](#), an entropy-augmented reward is adopted, which is given by

$$r_t^{\pi} = r_t + \alpha H(\pi(\cdot | s_t)), \quad (10)$$

where $\alpha \geq 0$ is the entropy coefficient and $H(\cdot)$ is the entropy function; that is, $H(p(x)) = -\sum_x p(x) \log p(x)$. By replacing r_t with r_t^{π} , the standard action- and state-value functions in Eqs. (1) are transformed into soft-value functions. According to [Haarnoja, Zhou, Abbeel, and Levine \(2018\)](#), the soft Q- and V-functions have the following relationships:

$$Q^{\pi}(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)} \{V^{\pi}(s')\} \quad (11a)$$

$$V^{\pi}(s) = \mathbb{E}_{a \sim \pi} \{Q^{\pi}(s, a) - \alpha \log \pi(a | s)\}. \quad (11b)$$

When $\alpha = 0$, (11a) and (11b) degenerate into (1a) and (1b), respectively.

Lemma 1. For arbitrary policies $\tilde{\pi}$ and π , the expected value function satisfies the following relationship

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{\tau \sim \tilde{\pi}} \left\{ \sum_{t=0}^{\infty} \gamma^t [A^{\pi}(s_t, a_t) + \alpha \log \frac{\pi(a_t | s_t)}{\tilde{\pi}(a_t | s_t)}] \right\}, \quad (12)$$

where $A^{\pi}(s_t, a_t) = Q^{\pi}(s_t, a_t) - V^{\pi}(s_t) - \alpha \log \pi(a_t | s_t)$ is the soft advantage function.

Proof. Please see [Appendix A](#). $\square$

Eq. (12) can be rewritten as the sum of the states instead of the time steps:

$$\begin{aligned} \eta(\tilde{\pi}) &= \eta(\pi) + \sum_s \sum_{t=0}^{\infty} \gamma^t P(s_t = s | \tilde{\pi}) \times \\ &\quad \sum_a \tilde{\pi}(a | s) [A^{\pi}(s, a) - \alpha \log \frac{\tilde{\pi}(a | s)}{\pi(a | s)}] \\ &= \eta(\pi) + \mathbb{E}_{s \sim \rho_{\tilde{\pi}}, a \sim \tilde{\pi}} \{A^{\pi}(s, a) - \alpha \log \frac{\tilde{\pi}(a | s)}{\pi(a | s)}\}. \end{aligned} \quad (13)$$

Obviously, if $\mathbb{E}_{s \sim \rho_{\tilde{\pi}}, a \sim \tilde{\pi}} \{A^{\pi}(s, a) - \alpha \log \frac{\tilde{\pi}(a | s)}{\pi(a | s)}\} \geq 0$ can be guaranteed, the target $\eta(\pi)$ will be improved. However, for a reason similar to that of TRPO, it is impractical to optimize Eq. (13).

Therefore, a tractable approximation objective is required. In addition, data collection is typically expensive in real-world scenarios. Hence, the objective should be to allow the use of off-policy data in its estimation. In addition, a multi-agent scenario is studied in which each agent aims to optimize its individual policies. Joint policy π is observed to be coupled with decentralized policies, indicating that each agent must access the parameters and gradients of the other agents' policies during the learning phase. However, despite employing the CTDE framework, reducing information sharing among agents remains desirable; thus, a decoupling joint policy is favored. Based on the above

reasons and inspired by the off-policy TRPO and on-policy MAPPO, this study proposes an approximation to $\eta(\tilde{\pi})$:

$$L_{\pi,b}(\tilde{\pi}) = \eta(\pi) + \sum_{i=1}^N \mathbb{E}_{s \sim \rho_b, a \sim \pi} \left\{ \frac{\tilde{\pi}^i(a^i|s)}{\pi^i(a^i|s)} A^\pi(s, a) \right\} - \alpha \mathbb{E}_{s \sim \rho_b} \{ D_{\text{KL}}(\tilde{\pi}(\cdot|s) \parallel \pi(\cdot|s)) \}, \quad (14)$$

where ρ_b is the discounted state distribution generated by an arbitrary behavior policy b . Eq. (14) indicates that the estimation of the approximated objective does not need to follow the on-policy state data generated by the current policy but can use an arbitrary behavior policy.

Policies in reinforcement learning are commonly represented by parameterized functions that can be easily handled, such as deep neural networks (DNNs). In this study, the current policy π^i for each agent i is parameterized as $\pi_{\theta^i}^i$, where θ^i represents a parameter vector. Similarly, policy $\tilde{\pi}^i$ is parameterized as $\tilde{\pi}_{\tilde{\theta}^i}^i$ with parameters $\tilde{\theta}^i$. Let $\theta = \{\theta^1, \dots, \theta^N\}$ and $\tilde{\theta} = \{\tilde{\theta}^1, \dots, \tilde{\theta}^N\}$. For brevity, we still use the symbols π_i and $\tilde{\pi}_i$, which implicitly refer to their respective parameterizations by θ_i and $\tilde{\theta}_i$. Using this parametric representation, we present the following proposition.

Proposition 1. Given the policies $\pi = \{\pi^1, \dots, \pi^N\}$, θ is updated according to

$$\theta' = \theta + \alpha_\theta \nabla_{\tilde{\theta}} L_{\pi,b}(\tilde{\pi})|_{\tilde{\theta}=\theta}, \quad (15)$$

where α_θ denotes the update size. If policies have a tabular representation (i.e., separate parameters for each state s), then there exists $\epsilon > 0$ such that for all $\alpha_\theta < \epsilon$,

$$\eta(\pi') \geq \eta(\pi). \quad (16)$$

Proof. The proof is provided in Appendix B. $\square$

Proposition 1 suggests that updating tabularly represented policies using the policy gradient of (14) can enhance the original target, given an appropriate update size. However, the application of tabularly represented policies is often impractical in real-world tasks because of their typically large and/or continuous state spaces. In such cases, where the tabular condition is not met, the policy improvement claim in Proposition 1 may not hold.² Specifically, if the policy parameters are relevant across different states, it remains uncertain whether updating π by using the gradient in (14) improves the original objective. To address this concern, Theorem 1 provides a more general relationship between $L_{\pi,b}(\tilde{\pi})$ and $\eta(\tilde{\pi})$ that does not rely on tabular policy assumptions.

Theorem 1. Define

$$M_{\pi,b}(\tilde{\pi}) = L_{\pi,b}(\tilde{\pi}) - \frac{2N\epsilon}{1-\gamma} \max_i \sqrt{D_{\text{KL}}^{\max}(\pi^i, \tilde{\pi}^i)} - \frac{2N\alpha}{(1-\gamma)} \max_i D_{\text{KL}}^{\max}(\tilde{\pi}^i, \pi^i). \quad (17)$$

Thus, $\eta(\tilde{\pi}) \geq M_{\pi,b}(\tilde{\pi})$.

Proof. The proof is provided in Appendix C. $\square$

When $\tilde{\pi} = \pi$, $D_{\text{KL}}^{\max}(\pi^i, \tilde{\pi}^i) = D_{\text{KL}}^{\max}(\tilde{\pi}^i, \pi^i) = 0$. Furthermore, $\mathbb{E}_{a \sim \pi} \{A^\pi(s, a)\} = 0$. Therefore, it can be obtained that $M_{\pi,b}(\pi) = \eta(\pi)$, so that $\eta(\tilde{\pi}) - \eta(\pi) \geq M_{\pi,b}(\tilde{\pi}) - M_{\pi,b}(\pi)$.

Therefore, if the lower bound $M_{\pi,b}(\tilde{\pi})$ is improved iteratively, then the original target $\eta(\tilde{\pi})$ is also improved monotonically. In other words, in each iteration round ($k = 0, 1, \dots$), if

$$\pi_{k+1} = \arg \max_{\tilde{\pi}} M_{\pi,b}(\tilde{\pi}), \quad (18)$$

² For the special case $b = \pi$; that is, in the on-policy case, the claim still holds even without satisfying the tabular condition

the original target must be non-decreasing; that is, $\eta(\pi_{k+1}) \geq \eta(\pi_k)$.

The objective of problem (18) is to maximize $L_{\pi,b}(\tilde{\pi})$ while minimizing the KL divergence. However, optimizing $M_{\pi,b}(\tilde{\pi})$ remains challenging because the maximum operator applies to the KL divergence over state s , resulting in an excessively large number of constraints. Therefore, similar to Schulman et al. (2015), the KL penalties in $M_{\pi,b}(\tilde{\pi})$ are transferred into constraints, and the maximum KL divergence is replaced by the averaged divergence over state s . Subsequently, the problem in (18) is transformed into

$$\max_{\tilde{\pi}} L_{\pi,b}(\tilde{\pi}) \quad (19a)$$

$$\text{s.t.} : \bar{D}_{\text{KL}}^{\rho_b}(\pi^i, \tilde{\pi}^i) \leq \delta_1^i, \forall i \in \mathcal{N} \quad (19b)$$

$$\bar{D}_{\text{KL}}^{\rho_b}(\tilde{\pi}^i, \pi^i) \leq \delta_2^i, \forall i \in \mathcal{N} \quad (19c)$$

where δ_1^i and δ_2^i denote the desired trust regions.

In this study, the squashed Gaussian policy (see Appendix D for introduction) is adopted in the experiments, which is the most common choice for continuous RL tasks (Andrychowicz et al., 2020). According to theorem 2 in Zhang, Liu, Chen, Wang, and Li (2021), when using a (squashed) Gaussian policy³, if $\bar{D}_{\text{KL}}^{\rho_b}(\pi^i, \tilde{\pi}^i) \leq \delta_1^i$, it holds that $\bar{D}_{\text{KL}}^{\rho_b}(\tilde{\pi}^i, \pi^i) \leq \delta_1^i + 2(\delta_1^i)^{1.5} + \mathcal{O}((\delta_1^i)^2) = \delta_2^i$. This observation suggests that ensuring (19b) guarantees constraint (19c). Therefore, to further simplify the problem (19), we neglect constraint (19c). For more general cases where quantitative analysis like Gaussian is challenging, empirically restraining $\bar{D}_{\text{KL}}^{\rho_b}(\pi^i, \tilde{\pi}^i)$ can still effectively restrain $\bar{D}_{\text{KL}}^{\rho_b}(\tilde{\pi}^i, \pi^i)$. Thus, this simplification remains reasonable from a practical perspective.

According to the additive property of KL divergence,

$$\begin{aligned} D_{\text{KL}}(\tilde{\pi}(\cdot|s) \parallel \pi(\cdot|s)) &= \sum_{i=1}^N D_{\text{KL}}(\tilde{\pi}^i(\cdot|s) \parallel \pi^i(\cdot|s)) \\ &= \sum_{i=1}^N \mathbb{E}_{a \sim \pi} \left\{ \frac{\tilde{\pi}^i(a^i|s)}{\pi^i(a^i|s)} \log \frac{\tilde{\pi}^i(a^i|s)}{\pi^i(a^i|s)} \right\}. \end{aligned} \quad (20)$$

Plugging Eq. (20) into $L_{\pi,b}(\tilde{\pi})$ and expanding $A^\pi(s, a)$, we obtain that

$$\begin{aligned} L_{\pi,b}(\tilde{\pi}) &= \eta(\pi) + \sum_{i=1}^N \mathbb{E}_{s \sim \rho_b, a \sim \pi} \left\{ \frac{\tilde{\pi}^i(a^i|s)}{\pi^i(a^i|s)} [Q^\pi(s, a) - V^\pi(s) - \alpha \log \pi^{-i}(a^{-i}|s) - \alpha \log \tilde{\pi}^i(a^i|s)] \right\}, \end{aligned} \quad (21)$$

where a^{-i} and π^{-i} indicate the joint actions and policies other than those of agent i , respectively.

Note that in Eq. (21), for a given π , the terms $\eta(\pi)$, $\mathbb{E}_{a \sim \pi} \left\{ \frac{\tilde{\pi}^i(a^i|s)}{\pi^i(a^i|s)} V^\pi(s) \right\}$ and $\mathbb{E}_{a \sim \pi} \left\{ \frac{\tilde{\pi}^i(a^i|s)}{\pi^i(a^i|s)} \log \pi^{-i}(a^{-i}|s) \right\}$ remain constant in each iteration round and can thus be omitted. Therefore, the objective $L_{\pi,b}(\tilde{\pi})$ is simplified as

$$\begin{aligned} L_{\pi,b}(\tilde{\pi}) &= \sum_{i=1}^N \mathbb{E}_{s \sim \rho_b, a \sim \pi} \left\{ \frac{\tilde{\pi}^i(a^i|s)}{\pi^i(a^i|s)} [Q^\pi(s, a) - \alpha \log \tilde{\pi}^i(a^i|s)] \right\}. \end{aligned} \quad (22)$$

With the above simplifications, the joint problem in (19) is decomposed into the following N subproblems for each agent $i \in \mathcal{N}$

$$\max_{\tilde{\pi}^i} L_{\pi,b}^i(\tilde{\pi}^i) \quad (23a)$$

$$\text{s.t.} : \bar{D}_{\text{KL}}^{\rho_b}(\pi^i, \tilde{\pi}^i) \leq \delta^i, \quad (23b)$$

where

$$\begin{aligned} L_{\pi,b}^i(\tilde{\pi}^i) &= \mathbb{E}_{s \sim \rho_b, a \sim \pi} \left\{ \frac{\tilde{\pi}^i(a^i|s)}{\pi^i(a^i|s)} [Q^\pi(s, a) - \alpha \log \tilde{\pi}^i(a^i|s)] \right\}, \end{aligned} \quad (24)$$

³ Theorem 2 in Zhang et al. (2021) originally applies to Gaussian policies. However, since squashing by $\tanh(\cdot)$ does not affect the KL divergence (Andrychowicz et al., 2020), this theorem also holds for squashed Gaussian policies.

δ^i is the desired KL constraint for agent i . It can be observed that in contrast to problem (9) (single-agent off-policy TRPO), problem (23) indicates that each agent i only needs to limit the update step between $\tilde{\pi}^i$ and π^i . However, it is unnecessary to consider the KL divergence between π and the behavior policy b .

Two popular existing methods can be applied to solve the problem (23). The first method is the approximated solution provided by TRPO. Although TRPO can reliably ensure the KL constraint, it requires taking second-order derivatives of the KL divergence and repeatedly computing the Fisher vector products for each update. Therefore, TRPO is computationally expensive, particularly in large and multi-agent domains. The second method is the clip mechanism proposed in PPO. PPO heuristically removes the KL constraint and restricts the update step by clipping the ratio of new and current policies. Although PPO requires only first-order gradients, this clip mechanism comes into effect only when the probability ratio has already exceeded the clip range. As a result, it cannot reliably limit the probability ratio within the clip range and therefore fails to constrain the KL divergence (Wang, He, & Tan, 2020).

Owing to these limitations, the two aforementioned methods cannot be adopted. This paper proposes a method that relies solely on first-order gradients (i.e., low computational complexity) and can effectively restrain the KL constraint. To achieve this objective, the following lemma is proposed:

Lemma 2. *The problem (23) is convex in $\tilde{\pi}^i$.*

Proof. In the objective (23a), $\mathbb{E}_{s \sim \rho_b, a \sim \pi} \left\{ \frac{\tilde{\pi}^i(a|s)}{\pi^i(a|s)} Q^\pi(s, a) \right\}$ is linear in $\tilde{\pi}^i$ and $-\mathbb{E}_{s \sim \rho_b, a \sim \pi} \left\{ \frac{\tilde{\pi}^i(a|s)}{\pi^i(a|s)} \alpha \log \tilde{\pi}^i(a|s) \right\} = \alpha \mathbb{E}_{s \sim \rho_b} \{ \mathcal{H}(\tilde{\pi}^i(\cdot|s)) \}$ is concave in $\tilde{\pi}^i$. The constraint (23b) is convex because of the convexity of the KL divergence. Therefore, the problem (23) is convex in $\tilde{\pi}^i$. $\square$

Owing to the convexity, the following Lagrangian function is constructed,

$$J(\tilde{\pi}^i, \beta^i) = L_{\pi, b}^i(\tilde{\pi}^i) - \beta^i (\bar{D}_{\text{KL}}^{\rho_b}(\pi^i, \tilde{\pi}^i) - \delta^i). \quad (25)$$

Subsequently, a dual gradient update procedure is applied to update $\tilde{\pi}^i$ and β^i alternately (Boyd, Boyd, & Vandenberghe, 2004). Specifically, for a given β^i , π_*^i is obtained by maximizing (25); that is,

$$\pi_*^i = \arg \max_{\tilde{\pi}^i} L_{\pi, b}^i(\tilde{\pi}^i) - \beta^i \bar{D}_{\text{KL}}^{\rho_b}(\pi^i, \tilde{\pi}^i) \quad (26)$$

With the given π_*^i , β^i is updated according to

$$\beta^i \leftarrow \beta^i - \lambda_{\beta^i} (\bar{D}_{\text{KL}}^{\rho_b}(\pi_*^i, \pi^i) - \delta^i), \quad (27)$$

where λ_{β^i} denotes the update size of β^i .

In practical implementations, it is unrealistic to fully optimize the Lagrangian function (25) with respect to the primal variables. However, a truncated version, that performs an incomplete optimization (even for a single gradient step) converges for convex problems (Boyd et al., 2004). It should be noted that the primal problem in (23) is convex in the policy space; however, practical implementations often parameterize the policy using nonlinear function approximators, such as neural networks, and optimize it in the parameter space rather than the policy space. Consequently, the convexity may not hold anymore. Nevertheless, the experimental findings indicate that this approach effectively constrains the KL divergence.

4. Practical implementation

In this section, based on problems (26) and (27), the practical algorithm is given. As mentioned in the previous section, π^i and $\tilde{\pi}^i$ are implicitly parameterized by θ^i and $\tilde{\theta}^i$, respectively. In addition, the centralized Q-function Q^π is parameterized by Q_ϕ with parameter vector ϕ .

First, we consider the estimator $L_{\pi, b}^i(\tilde{\pi}^i)$. A typical estimator of $L_{\pi, b}^i(\tilde{\pi}^i)$ is based on importance sampling, as in off-policy TRPO (Meng et al., 2022) and off-policy actor-critics (Degris, White, & Sutton, 2012). More clearly, $L_{\pi, b}^i(\tilde{\pi}^i)$ can be transformed into

$$\mathbb{E}_{s \sim \rho_b, a \sim b} \left\{ \frac{\tilde{\pi}^i(a|s)}{\pi^i(a|s)} \frac{\pi(a|s)}{b(a|s)} [Q^\pi(s, a) - \alpha \log \tilde{\pi}^i(a|s)] \right\}. \quad (28)$$

However, this estimator might result in high variance because the joint probability density ratio $\frac{\pi(a|s)}{b(a|s)}$ may be extremely large (even going to infinity), especially in continuous action and multi-agent cases.

Therefore, the reparameterization technique, which can provide a low-variance estimator and is simpler to implement, was adopted. To this end, Eq. (28) is first rewritten as

$$\mathbb{E}_{s \sim \rho_b, a^{-i} \sim \pi^{-i}, a^i \sim \tilde{\pi}^i} \{ Q^\pi(s, a^{-i}, a^i) - \alpha \log \tilde{\pi}^i(a^i|s) \}. \quad (29)$$

In reparameterization, the action of $\tilde{\pi}^i$ is reparameterized as

$$a^i = f_{\tilde{\theta}^i}(s; \epsilon), \quad (30)$$

where ϵ denotes the noise vector sampled from some fixed stochastic distribution $\mathcal{N}$. In this study, the squashed Gaussian policy is adopted, and a detailed introduction is provided in Appendix D. Then, $L_{\pi, b}^i(\tilde{\pi}^i)$ is reparameterized as

$$L_{\pi, b}^i(\tilde{\pi}^i) = \mathbb{E}_{s \sim \rho_b, a^{-i} \sim \pi^{-i}, \epsilon \sim \mathcal{N}} \{ Q^\pi(s, a^{-i}, f_{\tilde{\theta}^i}(s; \epsilon)) - \alpha \log \tilde{\pi}^i(f_{\tilde{\theta}^i}(s; \epsilon)|s) \}. \quad (31)$$

According to the chain rule, the gradient of $L_{\pi, b}^i(\tilde{\pi}^i)$ is given by Eq. (32), where Q^π is estimated by the tractable approximator Q_ϕ , and $\mathbb{E}_{s \sim \rho_b} \{ \cdot \}$ is replaced by $\mathbb{E}_{s \sim \mathcal{D}} \{ \cdot \}$. $\mathbb{E}_{s \sim \mathcal{D}} \{ \cdot \}$ is the sample average over state s from the replay buffer $\mathcal{D}$ that stores historical data because ρ_b can be an arbitrary distribution.

$$\begin{aligned} \nabla_{\tilde{\theta}^i} L_{\pi, b}^i(\tilde{\pi}^i) &= \mathbb{E}_{s \sim \mathcal{D}, a^{-i} \sim \pi^{-i}, \epsilon \sim \mathcal{N}} \left\{ \nabla_{\tilde{\theta}^i} f_{\tilde{\theta}^i}(s; \epsilon) [\nabla_{a^i} Q_\phi(s, a^{-i}, a^i) \right. \\ &\quad \left. - \alpha \nabla_{a^i} \log \tilde{\pi}^i(a^i|s)] \right|_{a^i = f_{\tilde{\theta}^i}(s; \epsilon)} - \alpha \nabla_{\tilde{\theta}^i} \log \tilde{\pi}^i(a^i|s) \Big|_{a^i = f_{\tilde{\theta}^i}(s; \epsilon)} \Big\}. \end{aligned} \quad (32)$$

It can be noticed that Eq. (32) is a general form of the MAD-DPG (Lowe et al., 2017) and can be seen as an extension of the deterministic policy to any tractable stochastic policy. More clearly, if the standard reward ($\alpha = 0$) and deterministic policy ($\epsilon = 0$) are applied, then the policy gradient (32) degenerates to that in the MADDPG. Therefore, our derivations can be regarded as a theoretical justification for MADDPG to somewhat extent.

By adding the gradient of the KL penalty, the total policy gradient is given by:

$$\begin{aligned} g(\tilde{\theta}^i) &= \nabla_{\tilde{\theta}^i} L_{\pi, b}^i(\tilde{\pi}^i) \\ &\quad - \beta^i \nabla_{\tilde{\theta}^i} \mathbb{E}_{s \sim \mathcal{D}} \{ D_{\text{KL}}(\pi^i(\cdot|s) \| \tilde{\pi}^i(\cdot|s)) \}. \end{aligned} \quad (33)$$

After the update of $\tilde{\pi}^i$, the parameters of π^i are updated according to $\theta^i \leftarrow \lambda_\theta \tilde{\theta}^i + (1 - \lambda_\theta) \theta^i$ with $\lambda_\theta \in (0, 1]$.

The gradient for updating the KL penalty coefficient β^i is

$$g(\beta^i) = \mathbb{E}_{s \sim \mathcal{D}} \{ D_{\text{KL}}(\pi^i(\cdot|s) \| \tilde{\pi}^i(\cdot|s)) \} - \delta^i. \quad (34)$$

The approximated Q-function Q_ϕ is updated by minimizing the following Huber loss:

$$J(\phi) = \hat{\mathbb{E}}_{(s, a, r, s') \sim \mathcal{D}} \{ \mathbf{H}_d(Q_\phi(s, a) - y_s) \}, \quad (35)$$

where the Huber loss function is defined as

$$\mathbf{H}_d(e) = \begin{cases} \frac{1}{2} e^2, & e \leq d, \\ d \cdot (|e| - \frac{1}{2} d), & \text{otherwise,} \end{cases} \quad (36)$$

and $y_s = r + \gamma(Q_\phi(s', a') - \alpha \log \pi(a'|s'))$, where $a' \sim \pi(\cdot|s')$. Q_ϕ is the target value function used to stabilize the learning process (Mnih et al., 2015). The parameters $\tilde{\phi}$ are updated from ϕ in the moving average, that is, $\tilde{\phi} = \lambda_\phi \phi + (1 - \lambda_\phi) \tilde{\phi}$ with $\lambda_\phi \in (0, 1]$. The Huber loss is a trade-off

between the mean squared error (MSE) and mean absolute error (MAE). It is more robust to singular data and performs well in experiments.

Practical Algorithm: The pseudocodes of the proposed off-policy multi-agent stochastic policy gradient (MASPG) are listed in Algorithm 1. To mitigate the overestimation problem of the Q function, two Q functions, Q_{ϕ_1} and Q_{ϕ_2} , are used, as in Fujimoto, Hoof, and Meger (2018). Therefore, y_s in Eq. (35) is calculated by $y_s = r + \gamma(\min_{k \in \{1,2\}} Q_{\phi_k}(s', a') - \alpha \log \pi(a'|s'))$, and $\nabla_{a^i} Q_{\phi}(s, a^{-i}, a^i)$ in Eq. (32) should be replaced with $\nabla_{a^i} \min_{k \in \{1,2\}} Q_{\phi_k}(s, a^{-i}, a^i)$. Some studies adopted a Q-function ensemble to further alleviate the overestimation problem and achieved excellent performance. However, the ensemble technique significantly increases the computation overheads so not been adopted. For example, Fujimoto, Hoof, and Meger (2022) trained an ensemble with ten Q-functions, which is much more computationally expensive than estimating two Q-functions.

Algorithm 1: Off-Policy MASPG

Initiate ϕ_k , for $k \in \{1, 2\}$, $\tilde{\theta}^i, \forall i$; $D \leftarrow \emptyset$
 $\bar{\phi}_k \leftarrow \phi_k$, for $k \in \{1, 2\}$, $\theta^i \leftarrow \tilde{\theta}^i, \forall i$.
for timestep $t = 0, 1, \dots, \infty$ **do**
 Each agent executes a_t^i according to $\pi^i(\cdot|s_t)$
 Receive the reward r_t
 Transfer to $s_{t+1} \sim P(\cdot|s_t, a_t)$
 $D \leftarrow D \cup \{s_t, a_t, r_t, s_{t+1}\}$
 for G rounds update **do**
 Randomly sample a mini-batch data from D
 Apply gradient ascent on $\tilde{\theta}^i$ using $g(\tilde{\theta}^i), \forall i$
 Apply gradient descent on β_i using $g(\beta_i), \forall i$
 Apply gradient descent on ϕ_k using $\nabla_{\phi_k} J(\phi_k), \forall k$
 $\theta^i \leftarrow \lambda_{\theta} \tilde{\theta}^i + (1 - \lambda_{\theta}) \theta^i, \forall i$
 $\bar{\phi}_k \leftarrow \lambda_{\phi} \phi_k + (1 - \lambda_{\phi}) \bar{\phi}_k, \forall k$
 end for
end for

5. Experiments

In this section, to validate the performance of the proposed off-policy MARL algorithm, we compare the proposed method with several state-of-the-art algorithms for representative and challenging tasks. The following subsections present the experimental settings, followed by a demonstration of the learning results.

5.1. Simulation setups

Multi-Agent MuJoCo (Peng et al., 2021): The proposed algorithms will be compared with other baseline methods on the benchmark, Multi-Agent MuJoCo (MaMuJoCo), which provides various challenging tasks of cooperative continuous multi-agent robotic control. MaMuJoCo was developed based on the popular single-agent robotic MuJoCo control suite (Todorov, Erez, & Tassa, 2012), which is included in OpenAI Gym (Brockman et al., 2016). MaMuJoCo separates a single robotic agent into several independent agents, each of which comprises one or more robot parts connected by controllable joints. These agents aim to learn the optimal motion cooperatively. For example, in the original MuJoCo in OpenAI Gym, task *Humanoid* has a 17-dimensional continuous action space. In MaMuJoCo, a single humanoid robot is partitioned into two agents, the lower and upper bodies, which have 9 and 8 dimensional continuous action spaces, respectively. The upper and lower bodies need to coordinately learn how to run quickly. For more detailed descriptions of MaMuJoCo please refer to Peng et al. (2021) and their open-source repository⁴.

Compared Algorithms: Our method will mainly be compared with the following baselines: (1) HATRPO (Kuba et al., 2022): An on-policy

Table 1

Hyperparameters.

Parameter	Value
Discount factor γ	0.99
Target update factor $\lambda_{\theta}, \lambda_{\phi}$	0.005
Learning rate for θ^i, ϕ, β^i	$\{3, 3, 1\} \times 10^{-4}$
Optimizer for θ^i, β^i, ϕ	Adam (Kingma & Ba, 2015)
Number of hidden layer	2
Units per hidden layer	256
Weighting decay factor	10^{-4}
Mini-batch size	256
Huber loss factor d	10
Update round G	1
Activation function	ReLU
Buffer size $ D $	10^6
Delta base δ_b	0.1, 0.005, 0.001
Episode limit	1000 (200 for <i>HumanoidStandup</i>)

Multi-Agent trust-region-based MARL algorithm developed recently. It has been claimed that HATRPO has achieved a new state-of-the-art (SOTA) performance on MaMuJoCo. (2) MAPPO (Yu et al., 2022): An MARL algorithm extended from the single-agent PPO, which achieved success in many cooperative multi-agent tasks. (3) MADDPG (Low et al., 2017): Multi-agent extension of DDPG. MADDPG in our implementation is slightly different from the original version. Because only cooperative tasks are considered, all agents share the same critic rather than separate ones as in the original implementation. Additionally, two Q-functions are adopted to stabilize the training process.

Hyperparameters: The default hyperparameters of our method are listed in Table 1. Policies and value functions are parameterized by multilayer perceptron (MLP). L2-regularizer (i.e., weighting decay) is adopted to relieve the overfitting. The delta base δ_b is the trust region constraint in one action dimension. Consider the *Humanoid* (9|8) task in MaMuJoCo, where the two agents have 9 and 8 dimensional continuous action spaces. If $\delta_b = 0.005$, the trust region constraint for the first agent is $\delta^1 = 9 \times 0.005$ and that for the second is $\delta^2 = 8 \times 0.005$. It is found that all three provided δ_b work well when an appropriate α is adopted. The episode limit is the maximum length of each episode during the training process. The entropy coefficient α is the only parameter that needs to be tuned across tasks because the reward scale varies in different scenarios. For *HumanoidStandup* (the task name in MaMuJoCo), α is set to 0.5. For *Walker2d*, $\alpha = 0.04$. For *Swimmer* and *HalfCheetah*, α is learned by using the automation-adjusting technique in Haarnoja, Zhou, Hartikainen, et al. (2018). For other tasks, $\alpha = 0.1$.

For MADDPG, the same settings listed in Table 1 are adopted, where the entropy coefficient α and terms related to β^i are removed because MADDPG uses the deterministic policy and has no trust region constraint. The exploration noise is set to 0.2. For HATRPO, the default hyperparameters in Kuba et al. (2022) are adopted, and for some tasks that they did not conduct experiments on (such as *Hopper*), the default setting in Kuba et al. (2022) for *Ant* is used. For MAPPO, the discount and GAE (generalized advantage estimation) factor are set to be 0.99 and 0.95, respectively. The PPO clip range is 0.2. The policy and value functions are parameterized using the two-layer ReLU-activated MLP with 256 units. The parameters of the neural networks are shared among agents, whereas the state (or observation) is appended with a one-hot agent index to allow the neural networks to distinguish the input of different agents. The batch size is 4000. The number of mini-batch is 16 (8 in *Ant*). The PPO epoch is 10 (8 in *Ant*). The learning rate is 4×10^{-4} .

5.2. Learning results

In this subsection, the learning results of our algorithm are compared with those of other baseline methods. Unless otherwise specified, all the experiments are conducted in a fully observable setting.

⁴ Code of MaMuJoCo is available at https://github.com/schroederdewitt/multiagent_mujoco

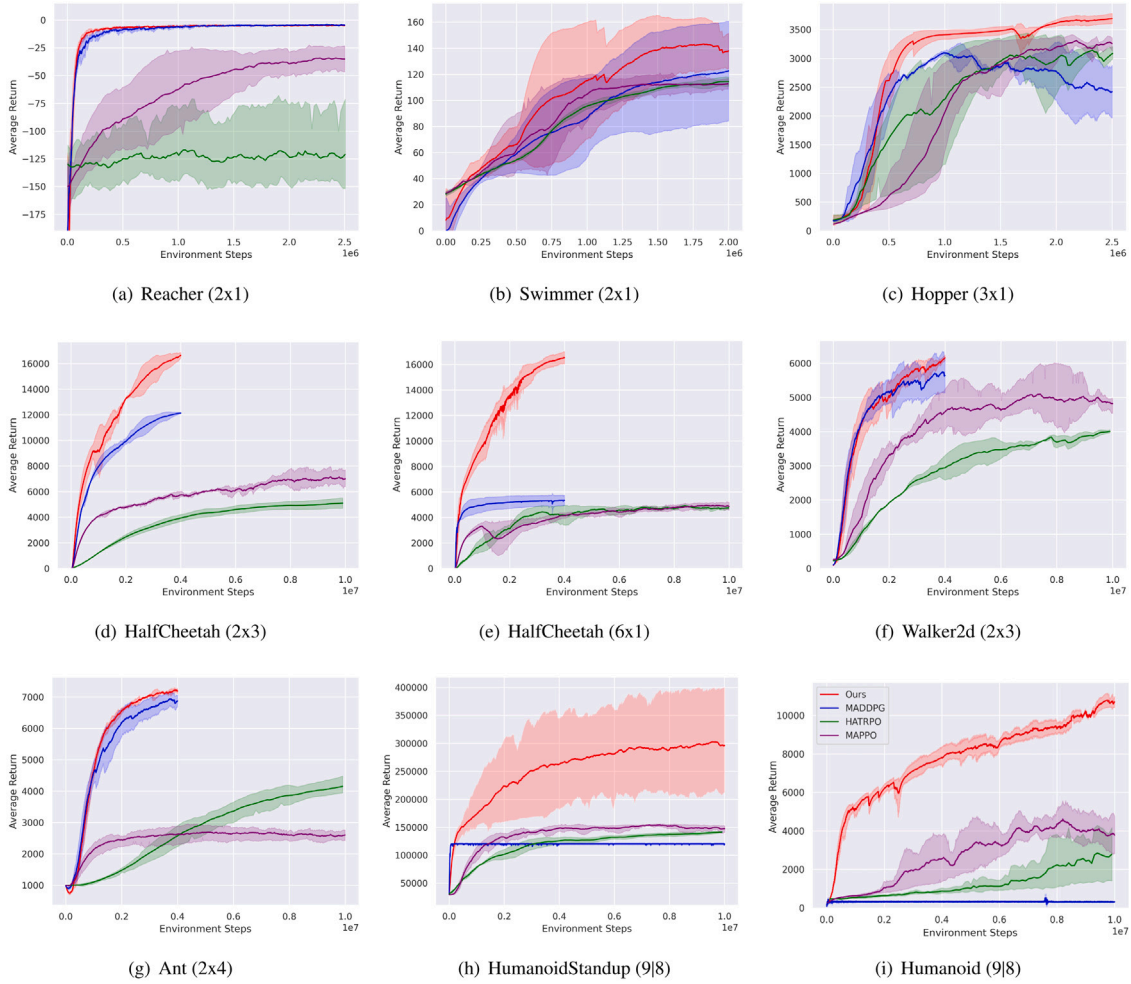


Fig. 1. The learning results of our algorithm compared with other MARL algorithms, including MAPPO, MADDPG and HATRPO.

Table 2

The average returns of both MARL algorithms with CTDE and Single-agent RL algorithms with fully centralized control. For each task, the highest score and those within 5% difference are bold.

	CTDE					Fully centralized control				
	Ours	MADDPG	MAPPO	HATRPO	HARL	Meta-SAC	GEM	AQE	ED2	T-PPO
Reacher	−3.87	−3.49	−34.13	−117.07	−4.9	/	/	/	/	−6.4
Swimmer	140.55	121.22	112.72	116.10	132.2	/	151.89	/	/	120.55
Hopper	3707	3217	3225	3187	3620	3408	3100	3541	3513	3510
HalfCheetah	16761	12170	7145	5199	8750	/	13190	13093	/	5188
Walker2d	6293	5814	5165	4008	6250	5738	5105	5517	6237	4210
Ant	7303	6996	2694	4282	6010	6521	5536	7345	6962	/
HumanoidStandup	300937	120498	155202	144082	/	/	/	/	/	/
Humanoid	10443	354.57	4613	2853	6750	9295	6013	8680	7304	7344

HATRPO is implemented using its open-source code. MADDPG and MAPPO are implemented with our fine-tuned codes, which consistently outperform (will be shown in Fig. 1 later) the results reported in Kuba et al. (2022) across most tasks. In some tasks, the two on-policy algorithms, HATRPO and MAPPO, are run for more environment steps.

The average return is estimated as $\frac{1}{K} \sum_{k=1}^K \sum_{t=1}^{T_{\max}} r_t^k$, where $K \geq 5$ represents the number of independent runs, $T_{\max} = 1000$ is the maximum episode length, and r_t^k denotes the reward at time step t in the k th round of experiments. Parentheses after the task name in each subcaption indicate the partition mode. For example, *Ant* (2 × 4) means that the robot in the *Ant* task is divided into 2 agents, each with a 4-dimensional continuous action space. Similarly, *Humanoid* (9|8) indicates the presence of two agents: one with a 9-dimensional continuous action space, and the other with an 8-dimensional continuous action space.

As illustrated in Fig. 1, our algorithm consistently outperforms all other compared methods (except for a tie with MADDPG on *Reacher* (2 × 1)), exhibiting significant performance advantages. Notably, the proposed method demonstrates exceptional capabilities for the most challenging tasks, namely, *Humanoid* (9|8) and *HumanoidStandup* (9|8), which involve a 17-dimensional action space. By comparing Figs. 1(d) and 1(e), where *HalfCheetah* is partitioned into 2 × 3 and 6 × 1 respectively, it is evident that the performance gaps between our method and its counterparts (especially MADDPG) increase as the number of agents increases. These results highlight the superiority of our algorithm in handling multi-agent tasks. In addition, the fine-tuned MADDPG achieves comparable performances on several tasks such as *Reacher*, *Walker2d*, and *Ant*. However, it still fails to perform well on more complex tasks such as *Humanoid* and *HumanoidStandup*. The experimental findings suggest that MADDPG quickly converges

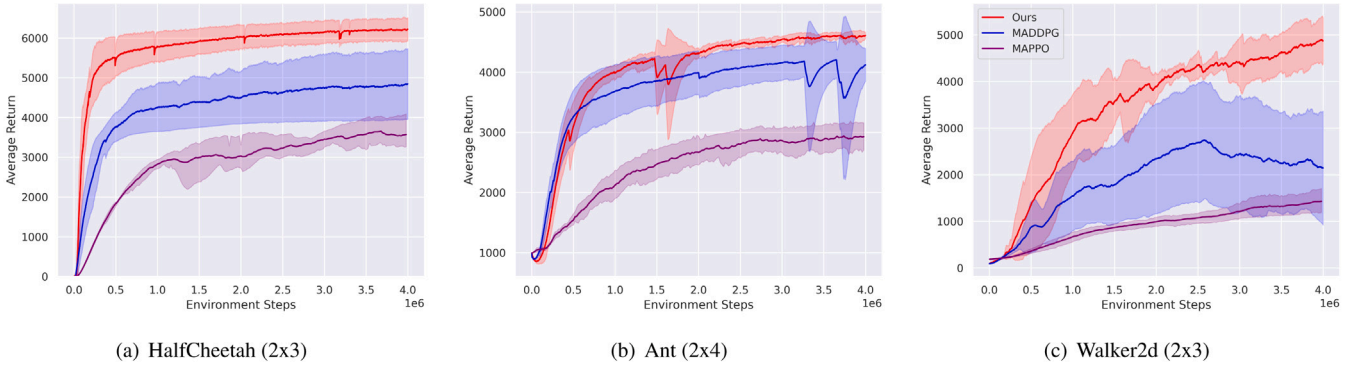


Fig. 2. The learning results of our algorithm compared with MAPPO and MADDPG on the partial observation setting.

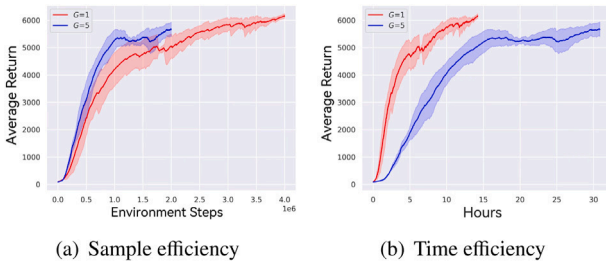


Fig. 3. The (a) sample efficiency and (b) time complexity with different update round G on the task *Walker2d* (2×3).

to a local optimum point in these two missions despite employing a large amount of exploration noise. This behavior is likely attributable to deterministic policies that easily fall into the saturation region of $\tanh(\cdot)$ within squashed Gaussian policies, leading to vanishing policy gradients.

In addition, we would like to further discuss the experiments conducted on the *HumanoidStandup* (9|8) task. This task aims to train a humanoid robot to stand up in a prone position, with the primary objective being the absolute height of the robot's head from the ground. The proposed algorithm was evaluated for this task through 15 independent runs, as indicated by the red shadows in Fig. 1(h). The experiments indicated that the proposed algorithm has encountered challenges related to local optima during this task. Specifically, although the best result achieved scores of up to 4×10^5 , there were instances in which the performance was as low as 2×10^5 . Overall, only 7/15 runs successfully learned how to stand. Despite the challenges faced by our algorithm to some extent, it still outperforms all other compared methods that can only achieve a sitting-up posture. To the best of our knowledge, this is the first successful application of a model-free method for training a humanoid robot to stand up in the *HumanoidStandup* task.

Comparisons with fully centralized control. In Table 2, to further validate the performance of our algorithm, our method is compared with an MARL method in CTDE framework, Heterogeneous-Agent Reinforcement Learning (HARL) (Zhong et al., 2023)⁵, and five recently

developed SOTA single-agent RL algorithms that implement fully centralized control, including meta soft actor-critic (Meta-SAC) (Wang & Ni, 2020), Generalizable Episodic Memory (GEM) (Hu, Ye, Zhu, Ren, & Zhang), Aggressive Q-Learning with Ensembles (AQE) (Fujimoto et al., 2022), Ensemble Deep Deterministic Policy Gradients (ED2) (Januszewski et al., 2021) and Truly PPO (T-PPO) (Wang, He, & Tan, 2020). The highest reported results from these studies were used. The slashes in the table indicate that the corresponding results were not provided in their papers. The results of task *HalfCheetah* under the CTDE framework refer only to the partition mode (2×3). For each task, the highest scores and those within a 5% difference are highlighted. Table 2 demonstrates that the proposed algorithm outperforms the other MARL algorithms under CTDE and single-agent RL with fully centralized control for almost all tasks. This further validates the effectiveness of the proposed method. Additionally, it can be observed that none of these methods have conducted experiments for *HumanoidStandup*, which is probably one of the most challenging tasks in (Ma)MuJoCo.

Comparisons in the partial observation setting. In Fig. 2, the proposed algorithm is compared with baseline MARL algorithms on three tasks in the partial observation setting. For all these tasks, $obsk = 2$, which is the configurable parameter in the code of MAMuJoCo used to adjust the extent of partial observation (see Appendix A in Peng et al. (2021) for more details). Because the code of HATRPO did not provide implementation for the partial observation setting, we only present the comparison results with MADDPG and MAPPO. Fig. 2 shows that in the partial observation setting, although the average returns are unsurprisingly reduced due to the absence of information, our proposed algorithm still achieves higher returns than other methods.

Sample and time efficiency. Fig. 3 illustrates the sample and time efficiency for different update rounds G for the *Walker2d* (2×3) task. Fig. 3(a) shows that increasing G significantly improves the sample efficiency, highlighting the advantage of off-policy implementation over on-policy algorithms. This is in contrast to the findings of an on-policy algorithm (Cobbe, Hilton, Klimov, & Schulman, 2021), where increasing the data reuse frequency during policy training actually leads to reduced performance. Notably, our algorithm achieves a score surpassing 5000 using only approximately 8×10^5 samples when $G = 5$, compared with approximately 2 million steps required when $G = 1$. However, after one million steps, the improvement rate for the case of $G = 5$ decreases and exhibits some fluctuations, possibly because high-frequency data reuse causes neural network overfitting. Fig. 3(b) shows the average return versus training time relationship. The time consumed per step is basically proportional to G , which is relatively expensive. In real-world scenarios, the training intensity can be adjusted based on practical requirements. For instance, in cases where data collection is costlier than computational devices (e.g., GPUs), a

⁵ HARL is a general framework introduced by Zhong et al. (2023), and there are several specific methods under the HARL. The highest scores across all these methods are listed in Table 2.

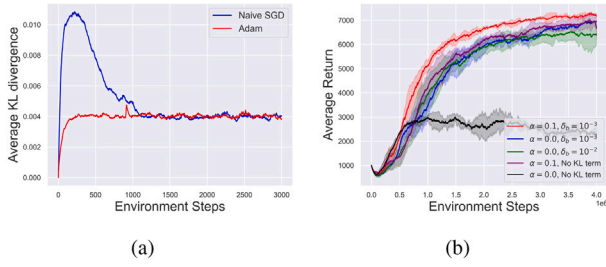


Fig. 4. (a) The average KL divergence with different optimizer, and (b) the impacts brought by the KL penalty with/without entropy augmentation.

larger value of G is recommended to enhance data efficiency and vice versa.

Impact of KL divergence. In Fig. 4, the impact of the KL penalty on the average return is studied, where the experiments are conducted on *Ant* (2×4). First, the effectiveness of the proposed method in constraining the KL divergence is verified. Fig. 4(a) presents the average KL divergence of the first agent's policy estimated by $\frac{1}{T_{\max}} \sum_{t=1}^{T_{\max}} D_{\text{KL}}(\pi^1(\cdot|s_t) \parallel \tilde{\pi}^1(\cdot|s_t))$, where Adam and naive stochastic gradient descent (SGD) optimizers are applied to update β^i . Here, the delta base $\delta_b = 0.001$; therefore, the target KL divergence is $4 \times 0.001 = 0.004$. The results demonstrate that the proposed method can effectively limit the average KL divergence to values close to the target value. The learning curves under different KL constraints are presented in Fig. 4(b). The legend “No KL term” indicates that no KL penalty term is applied ($\beta^i = 0, \forall i$). Without any KL constraint ($\alpha = 0$), the learning performance is significantly poor, even with a decrease in the average return after one million steps. However, when introducing the KL constraint, there is a substantial improvement in the average return, highlighting the necessity of restraining the policy update size. Interestingly, when entropy is added to the reward, the algorithm performs well without applying the KL penalty. This may be because maximizing the entropy of the policy also prevents aggressive update steps to some extent. Although the KL divergence and entropy are not necessarily correlated in theory, it is found in experiments (not shown in this paper) that increasing α can also effectively reduce the KL divergence. In the case with $\alpha = 0.1$ and $\delta_b = 0.001$, the algorithm achieves better performance than the case with $\alpha = 0.1$ and No KL term, which further validates the effectiveness of incorporating the KL penalty term.

6. Conclusions and further work

This study presents a novel off-policy algorithm for the MARL. Specifically, a more tractable approximation of the original MARL objective was developed that could effectively utilize off-policy data for the estimation. In addition, an entropy-augmented reward was introduced to enhance policy exploration. It has been proven that maximizing the proposed objective while constraining the KL divergence of policies guarantees a monotonic improvement in the original objective. Based on this monotonic improvement property, an off-policy MARL algorithm was proposed that safely updates the policy with the help of trust-region constraints. The proposed algorithm was tested on the MAMuJoCo benchmark, which provides various challenging tasks for continuous control. The experimental results demonstrated superior average returns compared with existing MARL algorithms under CTDE as well as state-of-the-art single-agent RL methods with fully centralized control, and exhibited reliable restraint of KL divergence.

In future work, a promising research direction is to improve the performance of the algorithm in handling tasks with a larger number of agents. With an increase in the number of agents, the multi-agent

credit assignment problem (see Foerster et al., 2018; Hu et al., 2022 for introduction) becomes severe. A promising scheme to address this issue is to apply the Shapley Value method to allocate the value function and introduce attention mechanisms to adapt the proposed algorithm to large-scale settings.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix A. Proof of Lemma 1

This proof follows the similar line in Schulman et al. (2015).

$$\begin{aligned}
 & \mathbb{E}_{\tau \sim \tilde{\pi}} \left\{ \sum_{t=0}^{\infty} \gamma^t [A^{\pi}(s_t, \mathbf{a}_t) + \alpha \log \frac{\pi(\mathbf{a}_t|s_t)}{\tilde{\pi}(\mathbf{a}_t|s_t)}] \right\} \\
 &= \mathbb{E}_{\tau \sim \tilde{\pi}} \left\{ \sum_{t=0}^{\infty} \gamma^t [Q^{\pi}(s_t, \mathbf{a}_t) - V^{\pi}(s_t) - \alpha \log \tilde{\pi}(\mathbf{a}_t|s_t)] \right\} \\
 &= \mathbb{E}_{\tau \sim \tilde{\pi}} \left\{ \sum_{t=0}^{\infty} \gamma^t [r_t + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s_t) - \alpha \log \tilde{\pi}(\mathbf{a}_t|s_t)] \right\} \\
 &= -\mathbb{E}_{\tau \sim \tilde{\pi}} \{V^{\pi}(s_0)\} + \mathbb{E}_{\tau \sim \tilde{\pi}} \left\{ \sum_{t=0}^{\infty} \gamma^t [r_t - \alpha \log \tilde{\pi}(\mathbf{a}_t|s_t)] \right\} \\
 &= -\eta(\pi) + \mathbb{E}_{\tau \sim \tilde{\pi}} \left\{ \sum_{t=0}^{\infty} \gamma^t [r_t + \alpha \mathcal{H}(\tilde{\pi}(\cdot|s_t))] \right\} \\
 &= -\eta(\pi) + \eta(\tilde{\pi})
 \end{aligned} \tag{A.1}$$

Rearranging, Lemma 1 follows. $\square$

Appendix B. Proof of Proposition 1

In this proof, the scaled entropy term is omitted for shortness, i.e., $\alpha = 0$. It can be easily proved that the results still hold when $\alpha > 0$. For each i , it has

$$\begin{aligned}
 & \nabla_{\tilde{\theta}} \mathbb{E}_{a \sim \pi} \left\{ \frac{\tilde{\pi}^i(a^i|s)}{\pi^i(a^i|s)} A^{\pi}(s, \mathbf{a}) \right\} \Big|_{\tilde{\theta}=\theta} \\
 &= \mathbb{E}_{a \sim \pi} \left\{ \frac{\nabla_{\tilde{\theta}} \tilde{\pi}^i(a^i|s)}{\pi^i(a^i|s)} [Q^{\pi}(s, \mathbf{a}) - V^{\pi}(s)] \right\} \Big|_{\tilde{\theta}=\theta} \\
 &\stackrel{(e)}{=} \mathbb{E}_{a \sim \pi} \left\{ \frac{\nabla_{\tilde{\theta}} \tilde{\pi}^i(a^i|s)}{\pi^i(a^i|s)} Q^{\pi}(s, \mathbf{a}) \right\} \Big|_{\tilde{\theta}=\theta} \\
 &= \mathbb{E}_{a \sim \pi} \left\{ \nabla_{\tilde{\theta}} \log \tilde{\pi}^i(a^i|s) Q^{\pi}(s, \mathbf{a}) \right\} \Big|_{\tilde{\theta}=\theta},
 \end{aligned} \tag{B.1}$$

where (e) holds due to $\mathbb{E}_{a \sim \pi} \left\{ \frac{\nabla_{\tilde{\theta}} \tilde{\pi}^i(a^i|s)}{\pi^i(a^i|s)} V^{\pi}(s) \right\} = 0$. Then, plugging (B.1) into $L_{\pi, b}(\tilde{\pi})|_{\tilde{\theta}=\theta}$,

$$\begin{aligned}
 & \nabla_{\tilde{\theta}} L_{\pi, b}(\tilde{\pi})|_{\tilde{\theta}=\theta} \\
 &= \sum_{i=1}^N \mathbb{E}_{s \sim \rho_b, a \sim \pi} \left\{ \nabla_{\tilde{\theta}} \log \tilde{\pi}^i(a^i|s) Q^{\pi}(s, \mathbf{a}) \right\} \Big|_{\tilde{\theta}=\theta} \\
 &= \mathbb{E}_{s \sim \rho_b, a \sim \pi} \left\{ \nabla_{\tilde{\theta}} \log \prod_{i=1}^N \tilde{\pi}^i(a^i|s) Q^{\pi}(s, \mathbf{a}) \right\} \Big|_{\tilde{\theta}=\theta} \\
 &= \mathbb{E}_{s \sim \rho_b, a \sim \pi} \left\{ \nabla_{\tilde{\theta}} \log \tilde{\pi}(\mathbf{a}|s) Q^{\pi}(s, \mathbf{a}) \right\} \Big|_{\tilde{\theta}=\theta} \\
 &= \mathbb{E}_{s \sim \rho_b} \left\{ \sum_a \nabla_{\tilde{\theta}} \tilde{\pi}(\mathbf{a}|s) Q^{\pi}(s, \mathbf{a}) \right\} \Big|_{\tilde{\theta}=\theta}.
 \end{aligned} \tag{B.2}$$

Recall that $\theta' = \theta + \alpha_{\theta} \nabla_{\tilde{\theta}} L_{\pi, b}(\tilde{\pi})|_{\tilde{\theta}=\theta}$. With tabular represented policies, there must exists an approximate α_{θ} such that for all s ,

$$\sum_a \pi(a|s) Q^{\pi}(s, \mathbf{a}) \leq \sum_a \pi'(a|s) Q^{\pi}(s, \mathbf{a}). \tag{B.3}$$

Repeatedly using the relationship in (B.3),

$$\begin{aligned}
V^\pi(s) &= \sum_a \pi(a|s) Q^\pi(s, a) = \sum_a \pi'(a|s) Q^\pi(s, a) \\
&= \sum_a \pi'(a|s) [r(s, a) + \gamma \sum_{s'} P(s'|s, a) V^\pi(s')] \\
&\leq \sum_a \pi'(a|s) [r(s, a) \\
&\quad + \gamma \sum_{s'} P(s'|s, a) \sum_{a'} \pi'(a'|s') Q^\pi(s', a')] \\
&\quad \vdots \\
&\leq \sum_a \pi'(a|s) Q^{\pi'}(s, a) \\
&= V^{\pi'}(s)
\end{aligned} \tag{B.4}$$

Since (B.4) holds for all $s \in S$, then $\eta(\pi') \geq \eta(\pi)$ holds. $\square$

Appendix C. Proof of Theorem 1

At the beginning of this proof, the following useful lemmas are given.

Lemma 3. For any two policy π_1 and π_2 , and an arbitrary bounded function $f : S \rightarrow \mathbb{R}$, it holds that

$$\begin{aligned}
&|\mathbb{E}_{s \sim \rho_{\pi_1}} \{f_1(s)\} - \mathbb{E}_{s \sim \rho_{\pi_2}} \{f_2(s)\}| \\
&\leq \mathbb{E}_{s \sim \rho_{\pi_1}} \{\max_s |f_1(s)|\} + \mathbb{E}_{s \sim \rho_{\pi_2}} \{\max_s |f_2(s)|\} \\
&= \max_s |f_1(s)| \mathbb{E}_{s \sim \rho_{\pi_1}} \{1\} + \max_s |f_2(s)| \mathbb{E}_{s \sim \rho_{\pi_2}} \{1\} \\
&\stackrel{(e)}{=} \frac{\max_s |f_1(s)| + \max_s |f_2(s)|}{1 - \gamma},
\end{aligned} \tag{C.1}$$

where the equality (e) in Eq. (C.1) holds due to $\mathbb{E}_{s \sim \rho_\pi} \{1\} = \sum_s \rho_\pi(s) = \sum_s \sum_{t=0}^{\infty} \gamma^t P(s_t = s | \pi) = \frac{1}{1-\gamma}$.

Lemma 4.

$$D_{KL}^{\max}(\pi^i, \tilde{\pi}^i) \geq 2D_{TV}^{\max}(\pi^i, \tilde{\pi}^i)^2 \tag{C.2}$$

$$D_{KL}^{\max}(\tilde{\pi}, \pi) \leq N \max_i D_{KL}^{\max}(\tilde{\pi}^i, \pi^i), \tag{C.3}$$

where the total variance distance is defined as $D_{TV}(\pi^i(\cdot|s) \parallel \tilde{\pi}^i(\cdot|s)) = \frac{1}{2} \sum_{a^i} |\tilde{\pi}^i(a^i|s) - \pi^i(a^i|s)|$, and $D_{TV}^{\max}(\pi^i, \tilde{\pi}^i) = \max_s D_{TV}(\pi^i(\cdot|s) \parallel \tilde{\pi}^i(\cdot|s))$.

Proof. For Eq. (C.2), according to Pollard (2009), it holds that $D_{KL}(\pi^i(\cdot|s) \parallel \tilde{\pi}^i(\cdot|s)) \geq 2D_{TV}(\pi^i(\cdot|s) \parallel \tilde{\pi}^i(\cdot|s))^2$. Taking the max operator on state s , the inequality still holds.

For Eq. (C.3), the KL divergence is additive for independent probability distributions, that is,

$$D_{KL}(\tilde{\pi}(\cdot|s), \pi(\cdot|s)) = \sum_{i=1}^N D_{KL}(\tilde{\pi}^i(\cdot|s) \parallel \pi^i(\cdot|s)). \tag{C.4}$$

Then taking maximum over s on both sides and further taking maximum on i on the right side yields Eq. (C.3). $\square$

Let us recall Eq. (13),

$$\begin{aligned}
\eta(\tilde{\pi}) &= \eta(\pi) + \mathbb{E}_{s \sim \rho_{\tilde{\pi}}, a \sim \tilde{\pi}} \{A^\pi(s, a) - \alpha \log \frac{\tilde{\pi}(a|s)}{\pi(a|s)}\} \\
&= \eta(\pi) + \mathbb{E}_{s \sim \rho_{\tilde{\pi}}, a \sim \tilde{\pi}} \{A^\pi(s, a) \\
&\quad - \alpha \mathbb{E}_{s \sim \rho_{\tilde{\pi}}} \{D_{KL}(\tilde{\pi}(\cdot|s) \parallel \pi(\cdot|s))\}\},
\end{aligned} \tag{C.5}$$

and Eq. (14),

$$\begin{aligned}
L_{\pi, b}(\tilde{\pi}) &= \eta(\pi) + \sum_{i=1}^N \mathbb{E}_{s \sim \rho_b, a \sim \pi} \left\{ \frac{\tilde{\pi}^i(a^i|s)}{\pi^i(a^i|s)} A^\pi(s, a) \right. \\
&\quad \left. - \alpha \mathbb{E}_{s \sim \rho_b} \{D_{KL}(\tilde{\pi}(\cdot|s) \parallel \pi(\cdot|s))\} \right\}.
\end{aligned} \tag{C.6}$$

The last terms on the right hand side (RHS) of Eq. (C.5) and (C.6) are first considered. According to Lemma 3, it can easily be obtained that

$$\begin{aligned}
&\mathbb{E}_{s \sim \rho_{\tilde{\pi}}} \{D_{KL}(\tilde{\pi}(\cdot|s) \parallel \pi(\cdot|s))\} \\
&\leq \mathbb{E}_{s \sim \rho_b} \{D_{KL}(\tilde{\pi}(\cdot|s) \parallel \pi(\cdot|s))\} + \frac{2D_{KL}^{\max}(\tilde{\pi}, \pi)}{(1-\gamma)}.
\end{aligned} \tag{C.7}$$

Then, the second terms on RHS of Eqs. (C.5) and (C.6). In Guo et al. (2020), it had been proved that

$$\begin{aligned}
&\mathbb{E}_{a \sim \tilde{\pi}} \{A^\pi(s, a)\} \leq 2N\epsilon \cdot \max_i D_{TV}^{\max}(\pi^i, \tilde{\pi}^i), \\
&\sum_{i=1}^N \mathbb{E}_{a \sim \pi} \left\{ \frac{\tilde{\pi}^i(a^i|s)}{\pi^i(a^i|s)} A^\pi(s, a) \right\} \leq 2N\epsilon \cdot \max_i D_{TV}^{\max}(\pi^i, \tilde{\pi}^i),
\end{aligned} \tag{C.8}$$

where $\epsilon = \max_{s, a} |A^\pi(s, a)|$. Using Lemma 3 and Eq. (C.8), it can be further obtained that

$$\begin{aligned}
\mathbb{E}_{s \sim \rho_{\tilde{\pi}}, a \sim \tilde{\pi}} \{A^\pi(s, a)\} &\geq \sum_{i=1}^N \mathbb{E}_{s \sim \rho_b, a \sim \pi} \left\{ \frac{\tilde{\pi}^i(a^i|s)}{\pi^i(a^i|s)} A^\pi(s, a) \right\} \\
&\quad - \frac{4N\epsilon}{1-\gamma} \max_i D_{TV}^{\max}(\pi^i, \tilde{\pi}^i)
\end{aligned} \tag{C.9}$$

By utilizing Eqs. (C.7) and (C.9), it is obtained that

$$\begin{aligned}
\eta(\tilde{\pi}) &\geq L_{\pi, b}(\tilde{\pi}) - \frac{4N\epsilon}{1-\gamma} \max_i D_{TV}^{\max}(\pi^i, \tilde{\pi}^i) \\
&\quad - \frac{2\alpha}{(1-\gamma)} D_{KL}^{\max}(\tilde{\pi}, \pi).
\end{aligned} \tag{C.10}$$

Finally, applying Lemma 4 on Eq. (C.10), it is obtained that

$$\begin{aligned}
\eta(\tilde{\pi}) &\geq L_{\pi, b}(\tilde{\pi}) - \frac{2N\epsilon}{1-\gamma} \max_i \sqrt{D_{KL}^{\max}(\pi^i, \tilde{\pi}^i)} \\
&\quad - \frac{2N\alpha}{(1-\gamma)} \max_i D_{KL}^{\max}(\tilde{\pi}^i, \pi^i).
\end{aligned} \tag{C.11}$$

This completes the proof of Theorem 1. $\square$

Appendix D. Squashed Gaussian policy

The squashed Gaussian policy is briefly introduced in this appendix. Let $\mu_{\theta^i}(s)$ and $\sigma_{\theta^i}(s)$ be the mean and standard variance vectors output by the neural network structure of the policy π^i . The random noise ϵ is sampled from the standard Normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The unbounded raw action is represented by $u^i = \mu_{\theta^i}(s) + \sigma_{\theta^i}(s) \odot \epsilon$, where $\odot$ denotes the element-wise product. Then, the unbounded action is squashed into $(-1, 1)$ by the hyperbolic tangent function

$$a^i = f_{\theta^i}(s; \epsilon) = \tanh(u^i). \tag{D.1}$$

The log-probability has the following form,

$$\log \pi^i(a^i|s) = \log v(u^i|s) - \sum_{k=1}^K \log(1 - \tanh^2(u_k^i)), \tag{D.2}$$

where v is the probability density function of normal distribution $\mathcal{N}(\mu_{\theta^i}(s), \sigma_{\theta^i}^2(s))$, and u_k^i is k th element of u^i .

It should be stressed that squashing does not affect the KL divergence (Andrychowicz et al., 2020), i.e.,

$$D_{KL}(\pi^i(\cdot|s) \parallel \tilde{\pi}^i(\cdot|s)) = D_{KL}(v^i(\cdot|s) \parallel \tilde{v}^i(\cdot|s)). \tag{D.3}$$

References

- Andrychowicz, M., Raichuk, A., Stańczyk, P., Orsini, M., Girgin, S., Marinier, R., et al. (2020). What matters in on-policy reinforcement learning? a large-scale empirical study. arXiv preprint arXiv:2006.05990.
- Boyd, S., Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.
- Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., et al. (2016). Openai gym. arXiv preprint arXiv:1606.01540.
- Cao, Y., Yu, W., Ren, W., & Chen, G. (2013). An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial Informatics*, 9(1), 427–438.

- Chen, W., Li, W., Liu, X., Yang, S., & Gao, Y. (2023). Learning explicit credit assignment for cooperative multi-agent reinforcement learning via polarization policy gradient. In *Proceedings of AAAI conference on artificial intelligence*.
- Cobbe, K. W., Hilton, J., Klimov, O., & Schulman, J. (2021). Phasic policy gradient. In *International conference on machine learning* (pp. 2020–2027).
- Degrís, T., White, M., & Sutton, R. (2012). Off-policy actor-critic. In *International conference on machine learning*.
- Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., & Whiteson, S. (2018). Counterfactual multi-agent policy gradients. In *Proceedings of AAAI conference on artificial intelligence* (pp. 2974–2982).
- Fujimoto, S., Hoof, H., & Meger, D. (2018). Addressing function approximation error in actor-critic methods. In *International conference on machine learning* (pp. 1587–1596).
- Fujimoto, S., Hoof, H., & Meger, D. (2022). Aggressive Q-learning with ensembles: Achieving both high sample efficiency and high asymptotic performance. In *Neural information processing systems*.
- Guo, D., Tang, L., Zhang, X., & Liang, Y.-C. (2020). Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 69(11), 13124–13138.
- Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning* (pp. 1861–1870).
- Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., et al. (2018). Soft actor-critic algorithms and applications. arXiv preprint arXiv:1812.05905.
- Hu, J., Hu, S., & Liao, S.-W. (2022). Policy regularization via noisy advantage values for cooperative multi-agent actor-critic methods. In *International foundation for autonomous agents and multiagent systems*.
- Hu, J., Jiang, S., Harding, S. A., Wu, H., & Liao, S.-W. (2021). Rethinking the implementation tricks and monotonicity constraint in cooperative multi-agent reinforcement learning. arXiv:2102.03479.
- Hu, H., Ye, J., Zhu, G., Ren, Z., & Zhang, C. Generalizable episodic memory for deep reinforcement learning. In *International conference on machine learning* (pp. 4380–4390).
- Hüttenrauch, M., Šošić, A., & Neumann, G. (2017). Guided deep reinforcement learning for swarm systems. arXiv preprint arXiv:1709.06011.
- Iqbal, S., & Sha, F. (2019). Actor-attention-critic for multi-agent reinforcement learning. In *International conference on machine learning* (pp. 2961–2970). PMLR.
- Januszewski, P., Olko, M., Królikowski, M., Świątkowski, J., Andrychowicz, M., Kuciński, Ł., et al. (2021). Continuous control with ensemble deep deterministic policy gradients. In *Deep RL workshop neurIPS 2021*.
- Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International conference on machine learning representations*.
- Kuba, J. G., Chen, R., Wen, M., Wen, Y., Sun, F., Wang, J., et al. (2022). Trust region policy optimisation in multi-agent reinforcement learning. In *International conference on learning representations*.
- Li, H., & He, H. (2020). Multi-Agent Trust Region policy optimization. arXiv preprint arXiv:2010.07916.
- Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., et al. (2016). Continuous control with deep reinforcement learning. In *Proceedings of international conference on learning representations*.
- Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994* (pp. 157–163). Elsevier.
- Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Pieter Abbeel, O., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proceedings of neural information processing systems* (pp. 6380–6391).
- Meng, W., Zheng, Q., Shi, Y., & Pan, G. (2022). An Off-Policy Trust Region policy optimization method with monotonic improvement guarantee for deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 33(5), 2223–2235.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533.
- Peng, B., Rashid, T., Schroeder de Witt, C., Kamienny, P.-A., Torr, P., Böhm, W., et al. (2021). Facmac: Factored multi-agent centralised policy gradients. *Advances in Neural Information Processing Systems (NIPS)*, 34, 12208–12221.
- Pollard, D. (2009). Asymptopia: an exposition of statistical asymptotic theory. <http://www.stat.yale.edu/pollard/Books/Asymptopia>.
- Rashid, T., Farquhar, G., Peng, B., & Whiteson, S. (2020). Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in Neural Information Processing Systems (NIPS)*, 33, 10199–10210.
- Rashid, T., Samvelyan, M., de Witt, C. S., et al. (2018). QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of international conference on machine learning*, vol. 10 (pp. 6846–6859).
- Samvelyan, M., Rashid, T., De Witt, C. S., Farquhar, G., Nardelli, N., Rudner, T. G., et al. (2019). The starcraft multi-agent challenge. In *Proceedings of international joint conference on autonomous agents and multiagent systems*, vol. 4 (pp. 2186–2188).
- Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust region policy optimization. In *International conference on machine learning* (pp. 1889–1897). PMLR.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.
- Shim, M. S., & Li, P. (2017). Biologically inspired reinforcement learning for mobile robot collision avoidance. In *2017 International joint conference on neural networks* (pp. 3098–3105). IEEE.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587), 484–489.
- Son, K., Kim, D., Kang, W. J., Hostallero, D. E., & Yi, Y. (2019). Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *Proceedings of international conference on machine learning* (pp. 10329–10346).
- Su, J., Adams, S., & Beling, P. (2021). Value-decomposition multi-agent actor-critics. In *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 13 (pp. 11352–11360).
- Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., et al. (2018). Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of international joint conference on autonomous agents and multiagent systems*, vol. 3 (pp. 2085–2087).
- Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the international conference on machine learning* (pp. 330–337).
- Todorov, E., Erez, T., & Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *IEEE/RSJ international conference on intelligent robots and systems*.
- Van der Pol, E., & Oliehoek, F. A. (2016). Coordinated deep reinforcement learners for traffic light control. In *30th Conference on neural information processing systems* (pp. 1–9).
- Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., et al. (2017). Sample efficient actor-critic with experience replay. In *International conference on learning representations*.
- Wang, T., Gupta, T., Mahajan, A., Peng, B., Whiteson, S., & Zhang, C. (2021). RODE: learning roles to decompose multi-agent tasks. In *Proceedings of the international conference on learning representations*.
- Wang, Y., Han, B., Wang, T., Dong, H., & Zhang, C. (2020). Dop: Off-policy multi-agent decomposed policy gradients. In *International conference on learning representations*.
- Wang, Y., He, H., & Tan, X. (2020). Truly proximal policy optimization. In *Uncertainty in artificial intelligence* (pp. 113–122).
- Wang, Y., & Ni, T. (2020). Meta-sac: Auto-tune the entropy temperature of soft actor-critic via metagradient. In *7th ICML workshop on automated machine learning*.
- Wang, J., Ren, Z., Liu, T., Yu, Y., & Zhang, C. (2020). QPLEX: Duplex dueling multi-agent Q-learning. In *Proceedings of international conference on learning representations*.
- Wang, J., Zhang, Y., Kim, T.-K., & Gu, Y. (2020). Shapley Q-value: A local reward approach to solve global reward games. In *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05 (pp. 7285–7292).
- Weisz, G., Budzianowski, P., Su, P.-H., & Gašić, M. (2018). Sample efficient deep reinforcement learning for dialogue systems with large action spaces. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(11), 2083–2097.
- Wen, Y., Chen, H., Yang, Y., Li, M., Tian, Z., Chen, X., et al. (2022). A game-theoretic approach to multi-agent trust region optimization. In *International conference on distributed artificial intelligence* (pp. 74–87). Springer.
- Wu, Y., Mansimov, E., Grosse, R. B., Liao, S., & Ba, J. (2017). Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. *Advances in Neural Information Processing Systems (NIPS)*, 30.
- Yang, Y., Hao, J., Liao, B., Shao, K., Chen, G., Liu, W., et al. (2020). Qatten: A general framework for cooperative multiagent reinforcement learning. arXiv preprint arXiv:2002.03939.
- Ye, D., Zhang, M., & Yang, Y. (2015). A multi-agent framework for packet routing in wireless sensor networks. *Sensors*, 15(5), 10026–10047.
- Yu, C., Velu, A., Vinitzky, E., Gao, J., Wang, Y., Bayen, A., et al. (2022). The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems (NIPS)*, 35, 24611–24624.
- Zhang, Y., Liu, W., Chen, Z., Wang, J., & Li, K. (2021). On the properties of Kullback-Leibler divergence between multivariate Gaussian distributions. arXiv preprint arXiv:2102.05485.
- Zhao, Q., Zhu, Y., Liu, Z., Wang, Z., & Chen, C. (2023). Boosting value decomposition via unit-wise attentive state representation for cooperative multi-agent reinforcement learning. arXiv preprint arXiv:2305.07182.
- Zhong, Y., Kuba, J. G., Hu, S., Ji, J., & Yang, Y. (2023). Heterogeneous-agent reinforcement learning. arXiv preprint arXiv:2304.09870.
- Zhou, M., Liu, Z., Sui, P., Li, Y., & Chung, Y. Y. (2020). Learning implicit credit assignment for cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems (NIPS)*, 33, 11853–11864.