Divergence-Regularized Multi-Agent Actor-Critic

Kefan Su[1] **Zongqing Lu**[1]

Abstract

Entropy regularization is a popular method in reinforcement learning (RL). Although it has many advantages, it alters the RL objective of the original Markov Decision Process (MDP). Though divergence regularization has been proposed to settle this problem, it cannot be trivially applied to cooperative multi-agent reinforcement learning (MARL). In this paper, we investigate divergence regularization in cooperative MARL and propose a *novel* off-policy cooperative MARL framework, divergence-regularized multi-agent actor-critic (DMAC). Theoretically, we derive the update rule of DMAC which is naturally off-policy and guarantees monotonic policy improvement and convergence in both the original MDP and divergence-regularized MDP. We also give a bound of the discrepancy between the converged policy and optimal policy in the original MDP. DMAC is a flexible framework and can be combined with many existing MARL algorithms. Empirically, we evaluate DMAC in a didactic stochastic game and StarCraft Multi-Agent Challenge and show that DMAC substantially improves the performance of existing MARL algorithms.

1. Introduction

Regularization is a common method for single-agent reinforcement learning (RL). The optimal policy learned by traditional RL algorithms is always deterministic (Sutton & Barto, 2018). This property may result in the inflexibility of the policy facing unknown environments (Yang et al., 2019). Entropy regularization is proposed to settle this problem by learning a policy according to the maximum-entropy principle (Haarnoja et al., 2017). Moreover, entropy regularization is beneficial to exploration and robustness for RL algorithms (Haarnoja et al., 2018). However, entropy

regularization is imperfect. Eysenbach & Levine (2019) pointed out that maximum-entropy RL modifies the original RL objective because of the entropy regularizer. Maximum-entropy RL is actually learning an optimal policy for the entropy-regularized Markov Decision Process (MDP) rather than the original MDP, *i.e.*, the converged policy may be biased. Nachum et al. (2017) analyzed a more general case for regularization in RL and proposed what we call *divergence regularization*. Divergence regularization is beneficial to exploration and may be helpful to the bias issue.

Regularization can also be applied to cooperative multi-agent reinforcement learning (MARL) (Agarwal et al., 2020; Zhang et al., 2021). However, most cooperative MARL algorithms do not use regularizers (Lowe et al., 2017; Foerster et al., 2018; Rashid et al., 2018; Son et al., 2019; Jiang et al., 2020; Wang et al., 2021a). Only few cooperative MARL algorithms such as FOP (Zhang et al., 2021) use entropy regularization, which may suffer from the drawback aforementioned. Divergence regularization, on the other hand, could potentially benefit cooperative MARL. In addition to its advantages mentioned above, divergence regularization can also help to control the step size of policy update which is similar to conservative policy iteration (Kakade & Langford, 2002) in single-agent RL. Conservative policy iteration and its successive methods such as TRPO (Schulman et al., 2015) and PPO (Schulman et al., 2017) can stabilize policy improvement (Touati et al., 2020). These methods use a surrogate objective for policy update, but the policies in centralized training with decentralized execution (CTDE) paradigm may not preserve the properties of the surrogate objective. Moreover, DAPO (Wang et al., 2019), a single-agent RL algorithm using divergence regularizer, cannot be trivially extended to cooperative MARL settings. Even with some tricks like V-trace (Espeholt et al., 2018) for off-policy correction, DAPO is essentially an on-policy algorithm and thus may not be sample-efficient in cooperative MARL settings.

In the paper, we propose divergence-regularized multi-agent actor-critic (DMAC), a *novel* off-policy cooperative MARL framework. We analyze the general iteration of DMAC and theoretically show that DMAC guarantees the monotonic policy improvement and convergence in both the original MDP and the divergence-regularized MDP. We also derive a bound of the discrepancy between the converged policy and

---

[1]School of Computer Science, Peking University. Correspondence to: Zongqing Lu <zongqing.lu@pku.edu.cn>.

the optimal policy in the original MDP. Besides, DMAC is beneficial to exploration and stable policy improvement by applying our update rule of target policy. We also propose and analyze *divergence policy iteration* in general cooperative MARL settings and a special case combined with value decomposition. Based on divergence policy iteration, we derive the off-policy update rule for the critic, policy, and target policy. Moreover, DMAC is a flexible framework and can be combined with many existing cooperative MARL algorithms to substantially improve their performance.

We empirically investigate DMAC in a didactic stochastic game and StarCraft Multi-Agent Challenge (Samvelyan et al., 2019). We combine DMAC with five representative MARL methods, *i.e.*, COMA (Foerster et al., 2018) for on-policy multi-agent policy gradient, MAAC (Iqbal & Sha, 2019) for off-policy multi-agent actor-critic, QMIX (Rashid et al., 2018) for value decomposition, DOP (Wang et al., 2021b) for the combination of value decomposition and policy gradient, and FOP (Zhang et al., 2021) for the combination of value decomposition and entropy regularization. Experimental results show that DMAC indeed induces better performance, faster convergence, and better stability in most tasks, which verifies the benefits of DMAC and demonstrates the advantages of divergence regularization over entropy regularization in cooperative MARL.

2. Related Work

MARL. MARL has been a hot topic in the field of RL. In this paper, we focus on cooperative MARL. Cooperative MARL is usually modeled as Dec-POMDP (Oliehoek et al., 2016), where all agents share a reward and aim to maximize the long-term return. Centralized training with decentralized execution (CTDE) (Lowe et al., 2017) paradigm is widely used in cooperative MARL. CTDE usually utilizes a centralized value function to address the non-stationary for multi-agent settings and decentralized policies for scalability. Many MARL algorithms adopt CTDE paradigm such as COMA, MAAC, QMIX, DOP, and FOP. COMA (Foerster et al., 2018) employs the counterfactual baseline which can reduce the variance as well as settle the credit assignment problem. MAAC (Iqbal & Sha, 2019) uses self-attention mechanism to integrate local observation and action of each agent and provides structured information for the centralized critic. Value decomposition (Sunehag et al., 2018; Rashid et al., 2018; Son et al., 2019; Yang et al., 2020; Wang et al., 2021a;b; Zhang et al., 2021; Sun et al., 2021; Pan et al., 2021; Peng et al., 2021) is a popular class of cooperative MARL algorithms. These methods express the global Q-function as a function of individual Q-functions to satisfy Individual-Global-Max (IGM), which means the optimal actions of individual Q-functions are corresponding to the optimal joint action of the global Q-function. QMIX

(Rashid et al., 2018) is a representative of value decomposition methods. It uses a hypernet to ensure the monotonicity of the global Q-function in terms of individual Q-functions, which is a sufficient condition of IGM. DOP (Wang et al., 2021b) is a method that combines value decomposition with policy gradient. DOP uses a linear value decomposition which is another sufficient condition of IGM and the linear value decomposition helps the compute of policy gradient. FOP (Zhang et al., 2021) is a method that combines value decomposition with entropy regularization and uses a more general condition, Individual-Global-Optimal, to replace IGM. In this paper, we will combine DMAC with these algorithms and show its improvement. MAPPO (Yu et al., 2021) is a CTDE version of PPO (Schulman et al., 2017), where a centralized state-value function is learned. However, the update rule of policy of MAPPO contradicts DMAC, so DMAC cannot be combined with MAPPO.

Regularization. Entropy regularization was first proposed in single-agent RL. Nachum et al. (2017) analyzed the entropy-regularized MDP and revealed the properties about the optimal policy and the corresponding Q-function and V-function. They also showed the equivalence of value-based methods and policy-based methods in entropy-regularized MDP. Haarnoja et al. (2018) pointed out maximum-entropy RL can achieve better exploration and stability facing the model and estimation error. Although entropy regularization has many advantages, Eysenbach & Levine (2019) showed entropy regularization modifies the MDP and results in the bias of the convergent policy. Yang et al. (2019) revealed the drawbacks of the convergent policy of general RL and maximum-entropy RL. The former is usually a deterministic policy (Sutton & Barto, 2018) that is not flexible enough for unknown situations, while the latter is a policy with non-zero probability for all actions which may be dangerous in some scenarios. Neu et al. (2017) analyzed the entropy regularization method from several views. They revealed a more general form of regularization which is actually divergence regularization and showed entropy regularization is just a special case of divergence regularization. Wang et al. (2019) absorbed previous results and proposed an on-policy algorithm, *i.e.*, DAPO. However, DAPO cannot be trivially applied to MARL. Moreover, its on-policy learning is not sample-efficient for MARL settings, and its off-policy correction trick V-trace (Espeholt et al., 2018) is also intractable in MARL. There are some previous studies in single-agent RL which use similar target policy to ours, but their purposes are quite different. Trust-PCL (Nachum et al., 2018) introduces a target policy as a trust-region constraint for maximum-entropy RL, but the policy is still biased by entropy regularizer. MIRL (Grau-Moya et al., 2019) uses a distribution that is only related to actions as the target policy to compute a mutual-information regularizer, but it still changes the objective of the original RL.

3. Preliminaries

Dec-POMDP is a general model for cooperative MARL. A Dec-POMDP is a tuple $M = \{S, A, P, Y, O, I, n, r, \gamma\}$. S is the state space, n is the number of agents, γ is the discount factor, and $I = \{1, 2 \cdots n\}$ is the set of all agents. $A = A_1 \times A_2 \times \cdots \times A_n$ represents the joint action space where A_i is the individual action space for agent i. $P(s'|s, \boldsymbol{a}) : S \times A \times S \to [0, 1]$ is the transition function, and $r(s, \boldsymbol{a}) : S \times A \to \mathbb{R}$ is the reward function of state s and joint action $\boldsymbol{a}$. Y is the observation space, and $O(s, i) : S \times I \to Y$ is a mapping from state to observation for each agent. The objective of Dec-POMDP is to maximize $J(\boldsymbol{\pi}) = \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{t=0} \gamma^t r(s_t, \boldsymbol{a}_t) \right]$, and thus we need to find the optimal joint policy $\boldsymbol{\pi}^* = \arg\max_{\boldsymbol{\pi}} J(\boldsymbol{\pi})$. To settle the partial observable problem, history $\tau_i \in \mathcal{T}_i = (Y \times A_i)^*$ is often used to replace observation $o_i \in Y$. As for policies in CTDE, each agent i has an individual policy $\pi_i(a_i|\tau_i)$ and the joint policy $\boldsymbol{\pi}$ is the product of each π_i. Though we calculate individual policy as $\pi_i(a_i|\tau_i)$ in practice, we will use $\pi_i(a_i|s)$ in analysis and proofs for simplicity.

Entropy regularization adds the logarithm of the probability of the sampled action according to current policy to the reward function. It modifies the optimization objective as $J_{\text{ent}}(\boldsymbol{\pi}) = \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{t=0} \gamma^t \left(r(s_t, \boldsymbol{a}_t) - \omega \log \boldsymbol{\pi}(\boldsymbol{a}_t|s_t) \right) \right]$. We also have the corresponding Q-function $Q_{\text{ent}}^{\boldsymbol{\pi}}(s, \boldsymbol{a}) = r(s, \boldsymbol{a}) + \gamma \mathbb{E} \left[V_{\text{ent}}^{\boldsymbol{\pi}}(s') \right]$ and V-function $V_{\text{ent}}^{\boldsymbol{\pi}}(s) = \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{t=0} \gamma^t \left(r(s_t, \boldsymbol{a}_t) - \omega \log \boldsymbol{\pi}(\boldsymbol{a}_t|s_t) \right) |s_0 = s \right]$. Given these definitions, we can deduce an interesting property $V_{\text{ent}}^{\boldsymbol{\pi}}(s) = \mathbb{E} \left[Q_{\text{ent}}^{\boldsymbol{\pi}}(s, \boldsymbol{a}) \right] + \omega \mathcal{H} \left(\boldsymbol{\pi}(\cdot|s) \right)$, where $\mathcal{H} \left(\boldsymbol{\pi}(\cdot|s) \right)$ represents the entropy of policy $\boldsymbol{\pi}(\cdot|s)$. $V_{\text{ent}}^{\boldsymbol{\pi}}(s)$ includes an entropy term which is the reason it is called *entropy regularization*.

4. Method

In this section, we first give the definition of divergence regularization. Then we propose the selection of target policy and derive its theoretical properties. Next we propose and analyze *divergence policy iteration*. Finally, we derive the update rules of the critic and actors and obtain the algorithm of DMAC. Note that all proofs are given in Appendix A.

4.1. Divergence Regularization

We maintain a target policy ρ_i for each agent i, which is different from the agent policy π_i. Together we have a joint target policy $\boldsymbol{\rho} = \prod_{i=1}^{n} \rho_i$. This joint target policy $\boldsymbol{\rho}$ modifies the objective function as $J_{\boldsymbol{\rho}}(\boldsymbol{\pi}) = \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{t=0} \gamma^t \left(r(s_t, \boldsymbol{a}_t) - \omega \log \frac{\boldsymbol{\pi}(\boldsymbol{a}_t|s_t)}{\boldsymbol{\rho}(\boldsymbol{a}_t|s_t)} \right) \right]$. That is, a regularizer $\log \frac{\boldsymbol{\pi}(\boldsymbol{a}_t|s_t)}{\boldsymbol{\rho}(\boldsymbol{a}_t|s_t)}$, which describes the *discrepancy* between policy $\boldsymbol{\pi}$ and target policy $\boldsymbol{\rho}$, is added to the reward function just like entropy regularization.

Given $\boldsymbol{\rho}$, we can define the corresponding V-function and Q-function for divergence regularization as follows,

$$V_{\boldsymbol{\rho}}^{\boldsymbol{\pi}}(s) = \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{t=0} \gamma^t (r(s_t, \boldsymbol{a}_t) - \omega \log \frac{\boldsymbol{\pi}(\boldsymbol{a}_t|s_t)}{\boldsymbol{\rho}(\boldsymbol{a}_t|s_t)}) |s_0 = s \right]$$

$$Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}}(s, \boldsymbol{a}) = r(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s' \sim P(\cdot|s, a)} \left[V_{\boldsymbol{\rho}}^{\boldsymbol{\pi}}(s') \right].$$

Further, by simple deduction, we have

$$V_{\boldsymbol{\rho}}^{\boldsymbol{\pi}}(s) = \mathbb{E}_{\boldsymbol{a} \sim \boldsymbol{\pi}(\cdot|s)} \left[Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}}(s, \boldsymbol{a}) - \omega \log \frac{\boldsymbol{\pi}(\boldsymbol{a}|s)}{\boldsymbol{\rho}(\boldsymbol{a}|s)} \right]$$

$$= \mathbb{E}_{\boldsymbol{a} \sim \boldsymbol{\pi}(\cdot|s)} \left[Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}}(s, \boldsymbol{a}) \right] - \omega D_{\text{KL}} \left(\boldsymbol{\pi}(\cdot|s) \| \boldsymbol{\rho}(\cdot|s) \right).$$

$V_{\boldsymbol{\rho}}^{\boldsymbol{\pi}}(s)$ includes an extra term which is the KL divergence between $\boldsymbol{\pi}$ and $\boldsymbol{\rho}$, and thus this regularizer is referred to as *divergence regularization*.

4.2. Target Policy

Intuitively, this regularizer $\log \frac{\boldsymbol{\pi}(\boldsymbol{a}_t|s_t)}{\boldsymbol{\rho}(\boldsymbol{a}_t|s_t)}$ could help to balance exploration and exploitation. For example, for some action $\boldsymbol{a}$, if $\boldsymbol{\rho}(\boldsymbol{a}|s) > \boldsymbol{\pi}(\boldsymbol{a}|s)$, then the regularizer is equivalent to adding a positive value to the reward and *vice versa*. Therefore, if we choose previous policy as target policy, the regularizer will encourage agents to take actions whose probability has decreased and discourage agents to take actions whose probability has increased. Additionally, the regularizer actually controls the discrepancy between current policy and previous policy, which could stabilize the policy improvement (Kakade & Langford, 2002; Schulman et al., 2015; 2017).

We further analyze the selection of target policy theoretically. Let $\mu_{\boldsymbol{\pi}}$ denote the state-action distribution given a policy $\boldsymbol{\pi}$. That is, $\mu_{\boldsymbol{\pi}}(s, \boldsymbol{a}) = d^{\boldsymbol{\pi}}(s) \boldsymbol{\pi}(\boldsymbol{a}|s)$, where $d^{\boldsymbol{\pi}}(s) = \sum_{t=0} \gamma^t \Pr(s_t = s | \boldsymbol{\pi})$ is the stationary distribution of states given $\boldsymbol{\pi}$. With $\mu_{\boldsymbol{\pi}}$, we can rewrite the optimization objective $J_{\boldsymbol{\rho}}(\boldsymbol{\pi})$ as follows,

$$J_{\boldsymbol{\rho}}(\boldsymbol{\pi}) = \sum_{s,\boldsymbol{a}} \mu_{\boldsymbol{\pi}}(s, \boldsymbol{a}) r(s, \boldsymbol{a}) - \omega \sum_{s,\boldsymbol{a}} \mu_{\boldsymbol{\pi}}(s, \boldsymbol{a}) \log \frac{\boldsymbol{\pi}(\boldsymbol{a}|s)}{\boldsymbol{\rho}(\boldsymbol{a}|s)}$$

$$= \sum_{s,\boldsymbol{a}} \mu_{\boldsymbol{\pi}}(s, \boldsymbol{a}) r(s, \boldsymbol{a}) - \omega D_{\text{C}} \left(\mu_{\boldsymbol{\pi}} \| \mu_{\boldsymbol{\rho}} \right), \quad (1)$$

where $D_{\text{C}} \left(\mu_{\boldsymbol{\pi}} \| \mu_{\boldsymbol{\rho}} \right) = \sum_{s,\boldsymbol{a}} \mu_{\boldsymbol{\pi}}(s, \boldsymbol{a}) \log \frac{\boldsymbol{\pi}(\boldsymbol{a}|s)}{\boldsymbol{\rho}(\boldsymbol{a}|s)}$ is a Bregman divergence (Neu et al., 2017). Therefore, the objective of the divergence-regularized MDP can be expressed as

$$\boldsymbol{\pi}_{\boldsymbol{\rho}}^* = \arg\max_{\boldsymbol{\pi}} \sum_{s,\boldsymbol{a}} \mu_{\boldsymbol{\pi}}(s, \boldsymbol{a}) r(s, \boldsymbol{a}) - \omega D_{\text{C}} \left(\mu_{\boldsymbol{\pi}} \| \mu_{\boldsymbol{\rho}} \right). \quad (2)$$

With this property, similar to Neu et al. (2017) and Wang et al. (2019), we can use the following iterative process,

$$\boldsymbol{\pi}^{t+1} = \arg\max_{\boldsymbol{\pi}} \sum_{s,\boldsymbol{a}} \mu_{\boldsymbol{\pi}}(s, \boldsymbol{a}) r(s, \boldsymbol{a}) - \omega D_{\text{C}} \left(\mu_{\boldsymbol{\pi}} \| \mu_{\boldsymbol{\pi}^t} \right).$$

$$(3)$$

In this iteration, by taking the target policy $\boldsymbol{\rho}$ as the previous policy $\boldsymbol{\pi}^t$ when updating the policy $\boldsymbol{\pi}^{t+1}$, we can obtain the following inequalities,

$$
\begin{aligned}
J(\boldsymbol{\pi}^{t+1}) &\geq J(\boldsymbol{\pi}^{t+1}) - \omega D_{\mathrm{C}}\left(\mu_{\boldsymbol{\pi}^{t+1}}\|\mu_{\boldsymbol{\pi}^t}\right) = J_{\boldsymbol{\pi}^t}(\boldsymbol{\pi}^{t+1}) \\
&\geq J(\boldsymbol{\pi}^t) - \omega D_{\mathrm{C}}\left(\mu_{\boldsymbol{\pi}^t}\|\mu_{\boldsymbol{\pi}^t}\right) \\
&= J(\boldsymbol{\pi}^t) \geq J_{\boldsymbol{\pi}^{t-1}}(\boldsymbol{\pi}^t).
\end{aligned}
$$

The first and the third inequalities are from $D_{\mathrm{C}}\left(\mu_{\boldsymbol{\pi}}\|\mu_{\boldsymbol{\rho}}\right) \geq 0$, and the second inequality is from the definition of $\boldsymbol{\pi}^{t+1}$. This indicates the policy sequence obtained by this iteration improves monotonically in both the divergence-regularized MDP (i.e., $J_{\boldsymbol{\pi}^t}(\boldsymbol{\pi}^{t+1}) \geq J_{\boldsymbol{\pi}^{t-1}}(\boldsymbol{\pi}^t)$) and the original MDP (i.e., $J(\boldsymbol{\pi}^{t+1}) \geq J(\boldsymbol{\pi}^t)$). Moreover, as the sequences $\{J_{\boldsymbol{\pi}^t}(\boldsymbol{\pi}^{t+1})\}$ and $\{J(\boldsymbol{\pi}^{t+1})\}$ are both bounded, these two sequences will converge. With these deductions, we can obtain the following proposition.

Proposition 1. *By iteratively applying the iteration* (3) *and taking* $\boldsymbol{\pi}^k$ *as* $\boldsymbol{\rho}^k$, *the policy sequence* $\{\boldsymbol{\pi}^k\}$ *converges and improves monotonically in both the divergence-regularized MDP and the original MDP.*

For the sake of simplicity, we denote the converged policy of iteration (3) and its corresponding Q-function and V-function as $\tilde{\boldsymbol{\pi}}^*$, $\tilde{Q}^*$ and $\tilde{V}^*$, respectively. Then we further discuss the property of $\tilde{\boldsymbol{\pi}}^*$, $\tilde{Q}^*$ and $\tilde{V}^*$. Actually, the expression of $\tilde{\boldsymbol{\pi}}^*$ is decided by the initial policy $\boldsymbol{\pi}^0$ and the action that obtains the optimal value of $\tilde{Q}^*$. We define the set containing the optimal actions as $U_s = \{\boldsymbol{a}' \in A | \boldsymbol{a}' = \arg\max_{\boldsymbol{a}} \tilde{Q}^*(s, \boldsymbol{a})\}$. Without loss of generality, we suppose that $\boldsymbol{\pi}^0(\boldsymbol{a}|s) > 0$ for all state-action pairs. Then we have the following proposition about $\tilde{\boldsymbol{\pi}}^*$.

Proposition 2. *All the probabilities of* $\tilde{\boldsymbol{\pi}}^*$ *lie in the optimal actions of* $\tilde{Q}^*$ *and are proportional to their probabilities in the initial policy* $\boldsymbol{\pi}^0$,

$$
\tilde{\boldsymbol{\pi}}^*(\boldsymbol{a}|s) = \mathbb{1}(\boldsymbol{a} \in U_s) \frac{\boldsymbol{\pi}^0(\boldsymbol{a}|s)}{\sum_{\boldsymbol{a}' \in U_s} \boldsymbol{\pi}^0(\boldsymbol{a}'|s)}.
$$

As for the property of the $\tilde{Q}^*$ and $\tilde{V}^*$, by the definition of $\tilde{Q}^*$ and $\tilde{V}^*$, we have the following proposition.

Proposition 3. $\tilde{Q}^*$ *is the same as* $Q^{\tilde{\boldsymbol{\pi}}^*}$, *the Q-function of the policy* $\tilde{\boldsymbol{\pi}}^*$ *in the original MDP, while* $\tilde{V}^*$ *is the same as* $V^{\tilde{\boldsymbol{\pi}}^*}$, *the V-function of the policy* $\tilde{\boldsymbol{\pi}}^*$ *in the original MDP.*

With all these results above, we can further derive the discrepancy between the policy $\tilde{\boldsymbol{\pi}}^*$ and the optimal policy $\boldsymbol{\pi}^*$ in the original MDP in terms of their V-functions, and have the following theorem.

Theorem 1. *If the initial policy* $\boldsymbol{\pi}^0$ *is a uniform distribution, then we have*

$$
\|V^{\tilde{\boldsymbol{\pi}}^*} - V^*\|_\infty \leq \frac{\omega}{1-\gamma} \log |A|, \tag{4}
$$

where V^* *is the optimal V-function in the original MDP and* $|A|$ *is the number of actions in* A.

This theorem tells us that the discrepancy between $\tilde{\boldsymbol{\pi}}^*$ and $\boldsymbol{\pi}^*$ can be controlled by the coefficient of the regularization term, ω.

At this stage, we can partly conclude the benefits of the divergence regularization or the target policy. The divergence regularization is beneficial to exploration which can be witnessed from the empirical results later. The policy sequence obtained by our selection of target policy converges and monotonically improves *not just in the regularized MDP, but also in the original MDP*. Moreover, the V-function of the converged policy in the *original* MDP can be sufficiently approximate to that of the optimal policy with a proper ω.

4.3. Divergence Policy Iteration

To complete the update in the iteration (3), we need to study the divergence-regularized MDP, given a fixed target policy $\boldsymbol{\rho}$.

From the perspective of policy evaluation, we can define an operator $\Gamma_{\boldsymbol{\rho}}^{\boldsymbol{\pi}}$ as

$$
\Gamma_{\boldsymbol{\rho}}^{\boldsymbol{\pi}} Q(s, \boldsymbol{a}) = r(s, \boldsymbol{a}) + \gamma \mathbb{E}\left[Q(s', \boldsymbol{a}') - \omega \log \frac{\boldsymbol{\pi}(\boldsymbol{a}'|s')}{\boldsymbol{\rho}(\boldsymbol{a}'|s')}\right]
$$

and have the following lemma.

Lemma 1 (Divergence Policy Evaluation). *For any initial Q-function* $Q^0(s, \boldsymbol{a}) : S \times A \to \mathbb{R}$, *we define a sequence* $\{Q^k\}$ *given operator* $\Gamma_{\boldsymbol{\rho}}^{\boldsymbol{\pi}}$ *as* $Q^{k+1} = \Gamma_{\boldsymbol{\rho}}^{\boldsymbol{\pi}} Q^k$. *Then, the sequence will converge to* $Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}}$ *as* $k \to \infty$.

After the evaluation of the policy, we need a way to improve the policy. We have the following lemma about policy improvement.

Lemma 2 (Divergence Policy Improvement). *If we define* $\boldsymbol{\pi}_{\mathrm{new}}$ *satisfying*

$$
\boldsymbol{\pi}_{\mathrm{new}}(\cdot|s) = \arg\min_{\boldsymbol{\pi}} D_{\mathrm{KL}}\left(\boldsymbol{\pi}(\cdot|s)\|\boldsymbol{u}(\cdot|s)\right), \tag{5}
$$

where $\boldsymbol{u}(\cdot|s) = \boldsymbol{\rho}(\cdot|s)\frac{\exp(Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\mathrm{old}}}(s, \cdot)/\omega)}{Z^{\boldsymbol{\pi}_{\mathrm{old}}}(s)}$ *and* $Z^{\boldsymbol{\pi}_{\mathrm{old}}}(s)$ *is a normalization term, then for all actions* $\boldsymbol{a}$ *and all states* s *we have* $Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\mathrm{new}}}(s, \boldsymbol{a}) \geq Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\mathrm{old}}}(s, \boldsymbol{a})$.

Lemma 1 and 2 can be seen as corollaries of the conclusion of Haarnoja et al. (2018). Lemma 2 indicates that given a policy $\boldsymbol{\pi}_{\mathrm{old}}$, if we find a policy $\boldsymbol{\pi}_{\mathrm{new}}$ according to (5), then the policy $\boldsymbol{\pi}_{\mathrm{new}}$ is better than $\boldsymbol{\pi}_{\mathrm{old}}$.

Lemma 2 does not make any assumption and is for general settings. Further, the policy improvement can be established and simplified based on value decomposition. In the following, we give an example for linear value decomposition

(LVD) like DOP (*i.e.*, $Q(s, \boldsymbol{a}) = \sum_i k_i(s)Q_i(s, a_i) + b(s)$) (Wang et al., 2021b).

Lemma 3 (**Divergence Policy Improvement with LVD**). *If Q-functions satisfy $Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}}(s, \boldsymbol{a}) = \sum_i k_i(s)Q_{\boldsymbol{\rho}}^{\pi_i}(s, a_i) + b(s)$ and we define π_{new}^i satisfying*

$$\pi_{\text{new}}^i(\cdot|s) = \arg\min_{\pi_i} D_{\text{KL}}\left(\pi_i(\cdot|s)\|u_i(\cdot|s)\right) \quad \forall i \in I,$$

where $u_i(\cdot|s) = \rho_i(\cdot|s)\frac{\exp\left(k_i(s)Q_{\boldsymbol{\rho}}^{\pi_{\text{old}}^i}(s, \cdot)/\omega\right)}{Z^{\pi_{\text{old}}^i}(s)}$ and $Z^{\pi_{\text{old}}^i}(s)$ is a normalization term, then for all actions $\boldsymbol{a}$ and all states s, we have $Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{new}}}(s, \boldsymbol{a}) \geq Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s, \boldsymbol{a})$.

Lemma 3 further tells us that if the MARL setting satisfies the condition of linear value decomposition, then each agent can optimize its individual policy with an objective of its own individual Q-function, which immediately improves the joint policy. By combining divergence policy evaluation and divergence policy improvement, we have the following theorem of *divergence policy iteration*.

Theorem 2 (**Divergence Policy Iteration**). *By iteratively using Divergence Policy Evaluation and Divergence Policy Improvement, we will get a sequence $\{Q^k\}$ and this sequence will converge to the optimal Q-function $Q_{\boldsymbol{\rho}}^*$ and the corresponding policy sequence will converge to the optimal policy $\boldsymbol{\pi}_{\boldsymbol{\rho}}^*$.*

Theorem 2 shows that with repeated application of divergence policy improvement and divergence policy evaluation, the policy can *monotonically* improve and converge to the optimal policy. We use $\boldsymbol{\pi}_{\boldsymbol{\rho}}^*$, $V_{\boldsymbol{\rho}}^*(s)$, and $Q_{\boldsymbol{\rho}}^*(s, \boldsymbol{a})$ to denote the optimal policy, Q-function, and V-function, respectively, given a target policy $\boldsymbol{\rho}$.

With all these results above, we have enough tools to obtain the practical update rule of the critic and actors of DMAC.

4.4. Divergence-Regularized Critic

For the update of the critic, we have the following proposition.

Proposition 4.

$$Q_{\boldsymbol{\rho}}^*(s, \boldsymbol{a}) = r(s, \boldsymbol{a}) + \gamma\mathbb{E}\left[Q_{\boldsymbol{\rho}}^*(s', \boldsymbol{a}') - \omega\log\frac{\boldsymbol{\pi}_{\boldsymbol{\rho}}^*(\boldsymbol{a}'|s')}{\boldsymbol{\rho}(\boldsymbol{a}'|s')}\right]$$

is tenable for all actions $\boldsymbol{a}' \in A$.

Proposition 4 gives an iterative formula for $Q_{\boldsymbol{\rho}}^*(s, \boldsymbol{a})$, with which we can have a loss function and update rule for learning the critic,

$$\mathcal{L}_Q = \mathbb{E}\left[(Q_\phi(s, \boldsymbol{a}) - y)^2\right],$$

where $y = r(s, \boldsymbol{a}) + \gamma\left(Q_{\tilde{\phi}}(s', \boldsymbol{a}') - \omega\log\frac{\boldsymbol{\pi}(\boldsymbol{a}'|s')}{\boldsymbol{\rho}(\boldsymbol{a}'|s')}\right),$

where ϕ and $\tilde{\phi}$ are respectively the weights of Q-function and target Q-function. The update of Q-function is similar to that in general MDP, except that the action for next state could be chosen *arbitrarily* while it must be the action that maximizes Q-function for next state in general MDP. This property greatly enhances the flexibility of learning Q-function, *e.g.*, we can easily extend it to TD(λ).

4.5. Divergence-Regularized Actors

DAPO (Wang et al., 2019) analyzes the divergence-regularized MDP from the perspective of policy gradient theorem (Sutton et al., 2000) and gives an on-policy update rule for single-agent RL. Unlike existing work, we focus on a different perspective and derive an off-policy update rule by taking into consideration the characteristics of MARL.

From Lemma 2, we can obtain an optimization target for policy improvement,

$$\arg\min_{\boldsymbol{\pi}} D_{\text{KL}}\left(\boldsymbol{\pi}(\cdot|s)\|\boldsymbol{\rho}(\cdot|s)\frac{\exp\left(Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s, \cdot)/\omega\right)}{Z^{\boldsymbol{\pi}_{\text{old}}}(s)}\right)$$

$$= \arg\max_{\boldsymbol{\pi}} \sum_{\boldsymbol{a}} \boldsymbol{\pi}(\boldsymbol{a}|s)\left(Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s, \boldsymbol{a}) - \omega\log\frac{\boldsymbol{\pi}(\boldsymbol{a}|s)}{\boldsymbol{\rho}(\boldsymbol{a}|s)}\right).$$

Then, we can define the objective of the actors,

$$\mathcal{L}_{\boldsymbol{\pi}} = \mathbb{E}_{s\sim\mathcal{D}}\left[\sum_{\boldsymbol{a}} \boldsymbol{\pi}(\boldsymbol{a}|s)\left(Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s, \boldsymbol{a}) - \omega\log\frac{\boldsymbol{\pi}(\boldsymbol{a}|s)}{\boldsymbol{\rho}(\boldsymbol{a}|s)}\right)\right],$$

where $\mathcal{D}$ is the replay buffer. Suppose each individual policy π_i has a corresponding parameterization θ_i. We can obtain the following policy gradient for each agent with some derivations given in Appendix A.9,

$$\nabla_{\theta_i}\mathcal{L}_{\boldsymbol{\pi}}$$

$$= \mathbb{E}\left[\nabla_{\theta_i}\log\pi_i(a_i|s)\left(Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s, \boldsymbol{a}) - \omega\log\frac{\boldsymbol{\pi}(\boldsymbol{a}|s)}{\boldsymbol{\rho}(\boldsymbol{a}|s)} - \omega\right)\right].$$

We need to point out that the key to *off-policy* update is that Lemma 2 does not limit the state distribution. It only requires the condition is satisfied for each state. Therefore, we can maintain a replay buffer to cover different states as much as possible, which is a common practice in off-policy learning. DAPO uses a similar formula to ours, but it obtains the formula from policy gradient theorem, which requires the state distribution of the current policy.

Further, we can add a counterfactual baseline to the gradient. First, we have the following equation about the counterfactual baseline (Foerster et al., 2018),

$$\mathbb{E}_{s\sim\mathcal{D},\boldsymbol{a}\sim\boldsymbol{\pi}}\left[\nabla_{\theta_i}\log\pi_i(a_i|s)b(s, a_{-i})\right] = 0,$$

where a_{-i} denotes the joint action of all agents except agent i. Next, we take the baseline as

$$b(s, a_{-i}) = \mathbb{E}_{a_i\sim\pi_i}[(Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s, \boldsymbol{a}) - \omega\log\frac{\boldsymbol{\pi}(\boldsymbol{a}|s)}{\boldsymbol{\rho}(\boldsymbol{a}|s)} - \omega)].$$
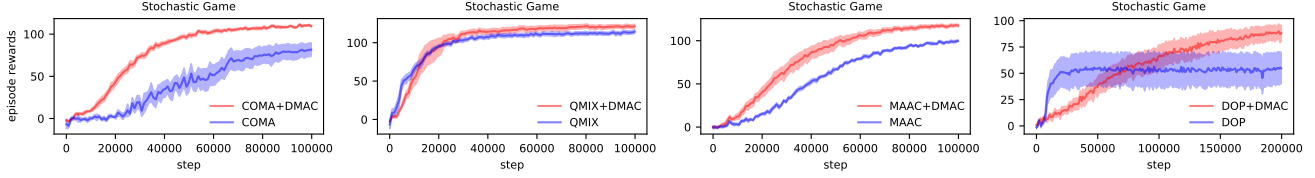
Figure 1. Learning curves in terms of episode rewards of COMA, MAAC, QMIX and DOP groups in randomly generated stochastic game.

Then, the gradient for each agent i can be modified as follows,

$$\nabla_{\theta_i}\mathcal{L}_{\boldsymbol{\pi}} = \mathbb{E}\Bigg[\nabla_{\theta_i}\log\pi_i(a_i|s)(Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s,\boldsymbol{a}) - \omega\log\frac{\pi_i(a_i|s)}{\rho_i(a_i|s)}$$
$$- \mathbb{E}_{a_i\sim\pi_i}[Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s,\boldsymbol{a})] + \omega D_{\text{KL}}(\pi_i(\cdot|s)\|\rho_i(\cdot|s)))\Bigg].$$

In addition to variance reduction and credit assignment, this counterfactual baseline eliminates the policies of other agents from the gradient. This property makes it convenient to calculate the gradient and easy to select the target policy for each agent. Moreover, if the linear value decomposition condition is satisfied, we have the following gradient formula,

$$\nabla_{\theta_i}\mathcal{L}_{\boldsymbol{\pi}} = \mathbb{E}\Bigg[\nabla_{\theta_i}\log\pi_i(a_i|s)\big(k_i(s)A_{\boldsymbol{\rho}}^{\pi_{\text{old}}^i}(s,a_i)$$
$$- \omega\log\frac{\pi_i(a_i|s)}{\rho_i(a_i|s)} + \omega D_{\text{KL}}(\pi_i(\cdot|s)\|\rho_i(\cdot|s)))\Bigg],$$

where $A_{\boldsymbol{\rho}}^{\pi_{\text{old}}^i}(s,a_i) = Q_{\boldsymbol{\rho}}^{\pi_{\text{old}}^i}(s,a_i) - \mathbb{E}_{\tilde{a}_i\sim\pi_i}\left[Q_{\boldsymbol{\rho}}^{\pi_{\text{old}}^i}(s,\tilde{a}_i)\right]$.

4.6. Algorithm

Every iteration of (3) needs a converged policy. However, it is intractable to perform this update rule precisely in practice. Thus, we propose an alternative approximate method. For each agent, we update the policy π_i and the target policy ρ_i respectively as $\theta_i = \theta_i + \beta\nabla_{\theta_i}\mathcal{L}_{\boldsymbol{\pi}}$ and $\tilde{\theta}_i = (1-\tau)\tilde{\theta}_i + \tau\theta_i$, where β is the learning rate, $\tilde{\theta}_i$ is the weights of ρ_i, and τ is the hyperparameter for soft update. Here we use one gradient step to replace the max operator in (3). From Theorem 2 and previous discussion, we know that optimizing $\mathcal{L}_{\boldsymbol{\pi}}$ can maximize $J_{\boldsymbol{\rho}}(\boldsymbol{\pi})$, so we use $\nabla_{\theta_i}\mathcal{L}_{\boldsymbol{\pi}}$ in the gradient step for off-policy training instead of the gradient step directly optimizing $J_{\boldsymbol{\rho}}(\boldsymbol{\pi})$ in (1).

Moreover, even if we take the target policies $\{\rho^k\}$ as the moving average of the policies $\{\boldsymbol{\pi}^k\}$, the properties such as monotonic improvements, convergences of value functions and policies, and the bound of the biases, will be still conserved. The details of these results are included in Appendix A.10.

Now we have all the update rules of DMAC. The training of DMAC is a typical off-policy learning process, which is given in Appendix B for completeness.

5. Experiments

In this section, we first empirically study the benefits of DMAC and investigate how DMAC improves the performance of existing MARL algorithms in a didactic stochastic game and five SMAC maps. Then, we demonstrate the advantages of divergence regularizer over entropy regularizer.

5.1. Improvements of Existing Methods

DMAC is a flexible framework and can be combined with many existing MARL algorithms. In the experiments, we choose four representative algorithms for different types of methods: COMA (Foerster et al., 2018) for on-policy multi-agent policy gradients, MAAC (Iqbal & Sha, 2019) for off-policy multi-agent actor-critic, QMIX (Rashid et al., 2018) for value decomposition, DOP (Wang et al., 2021b) for the combination of value decomposition and policy gradient. These algorithms need minor modifications to fit the framework of DMAC. We denote these modified algorithms as COMA+DMAC, MAAC+DMAC, QMIX+DMAC, and DOP+DMAC. Generally, our modification is limited and tries to keep the original architecture so as to fairly demonstrate the improvement of DMAC. The details of the modifications and hyperparameters are included in Appendix C. All the curves in our plots correspond to the mean value of five training runs with different random seeds, and shaded regions indicate 95% confidence interval.

A DIDACTIC EXAMPLE

We first test the four groups of methods in a stochastic game where agents share the reward. The stochastic game is generated randomly for the reward function and transition probabilities with 30 states, 3 agents and 5 actions for each agent. Each episode contains 30 timesteps in this game.
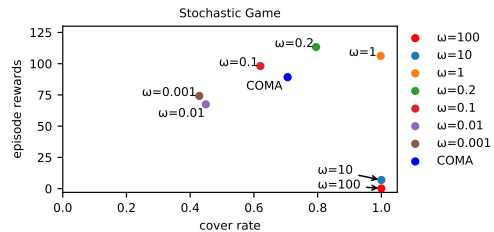

Figure 2. The scatter plot of the converged COMA and COMA+DMAC with different ω in terms of the episode rewards and cover rate in the randomly generated stochastic game.
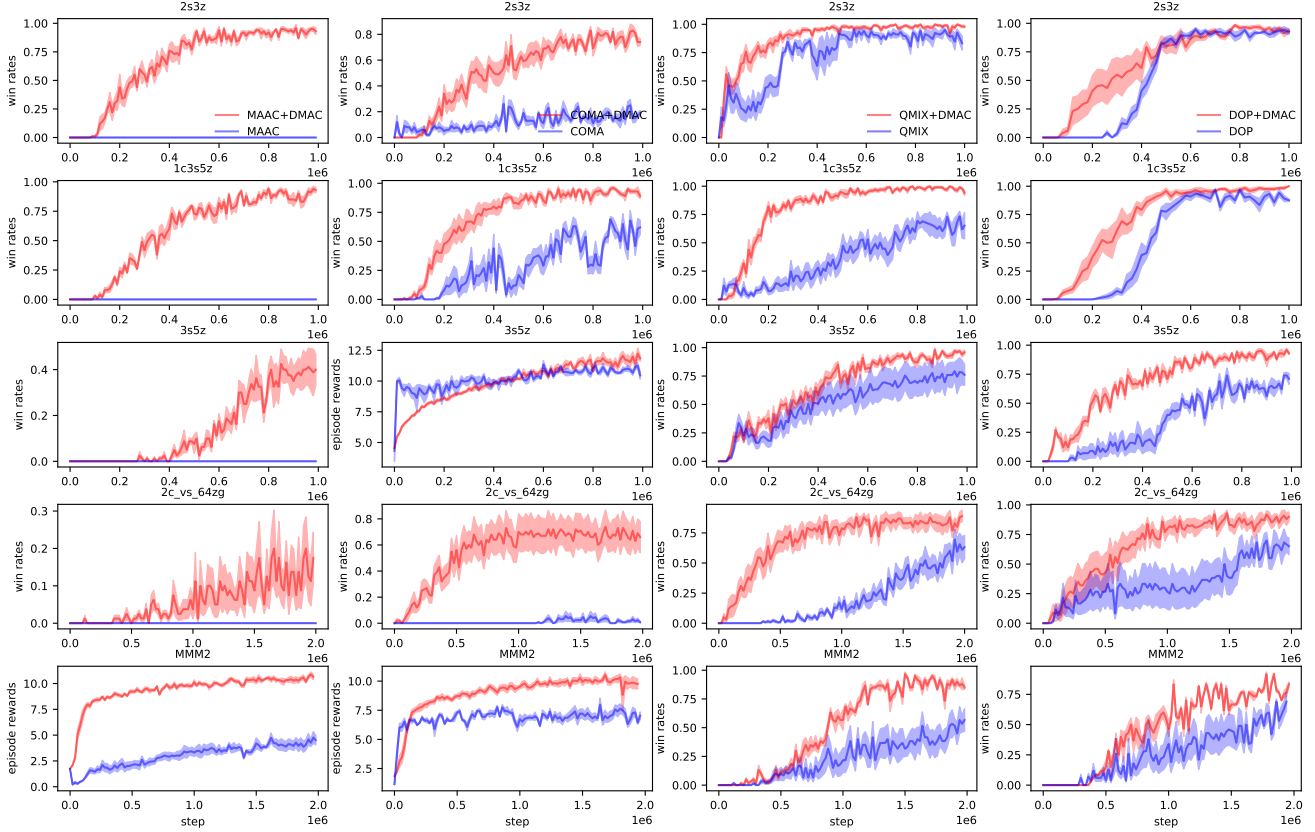
Figure 3. Learning curves in terms of win rates or episode rewards of COMA, MAAC, QMIX and DOP groups in five SMAC maps (each row corresponds to a map and each column corresponds to a group).

The performance of these methods is illustrated in Figure 1. We can find that DMAC performs better than the baseline at the end of the training in all the four groups. Moreover, COMA+DMAC, QMIX+DMAC and MAAC+DMAC learn faster than their baselines. Though DOP learns faster than DOP+DMAC at the start, it falls into a sub-optimal policy and DOP+DMAC finds a better policy in the end.

We also show the benefit of exploration in this stochastic game for the convenience of statistics. We evaluate the exploration in terms of the *cover rate* of all state-action pairs, *i.e.*, the ratio of the explored state-action pairs to all state-action pairs. The cover rates of COMA and COMA+DMAC with different ω are illustrated in Figure 2. We first study the effect of ω for the tradeoff between exploration and exploitation. From our previous analysis, a smaller ω gives a tighter bound for the converged performance of DMAC, but it also means that the benefits of the regularizer for exploration is smaller which will also affect the performance. So we need to choose a proper ω in practice. We can find in Figure 2 that when ω is large such as the case $\omega = 10$ and $\omega = 100$, the exploration is guaranteed but the performance is quite low since the regularizer is much larger than the environmental reward, and when ω is small such as the case $\omega = 0.001$ and $\omega = 0.01$, the performance is limited by

the ability of exploration. Only when ω lies in a proper interval such as $\omega = 0.1$, $\omega = 0.2$ and $\omega = 1$, the agents of DMAC can obtain a good balance between exploration and exploitation in this task.

We use COMA here as a representation of the traditional policy gradient method in cooperative MARL. In practice, we choose the $\omega = 0.2$ for COMA+DMAC. We can find that the cover rate of COMA+DMAC ($\omega = 0.2$) is higher than COMA, which can be an evidence for the benefit of exploration of DMAC. The cover rates of other three groups of algorithms and the learning curves for methods in Figure 2 are available in Appendix D.

SMAC

We test all the methods in five tasks of SMAC (Samvelyan et al., 2019). The introduction of the SMAC environment and the training details are included in Appendix C. The learning curves in terms of win rate (or episode rewards) of all the methods in the five SMAC maps are illustrated in Figure 3 (four columns for four groups of algorithms and five rows for five different maps in SMAC). For the case that both the baseline and DMAC can hardly win such as 3s5z and MMM2 for the COMA group and MMM2 for the MAAC group, we use the episode rewards to show the dif-
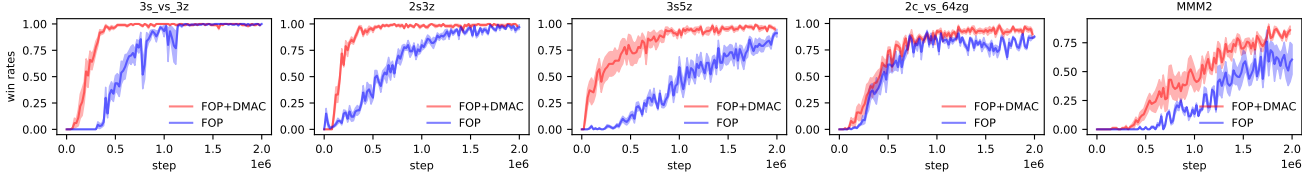
Figure 4. Learning curves in terms of win rates of FOP and FOP+DMAC in five SMAC maps.

ference. In addition, the learning curves in terms of episode rewards are available in Appendix D, including two additional maps, 3m and 8m. We show the empirical result of DAPO in the map of 3m in the first figure of the second column in Figure 8 in Appendix D. It can be seen that DAPO cannot obtain a good performance in the simple task of SMAC, so we skip it in other SMAC tasks. The reason for the low performance of DAPO may be that DAPO omits the correction for $d_\pi(s)/d_{\pi_t}(s)$ in policy update which introduces bias in the gradient of policy, and uses V-trace as off-policy correction which however is biased. These drawbacks may be magnified in MARL settings. The superiority of our naturally off-policy method over the biased off-policy correction method can be partly seen from the large performance gap between COMA+DMAC and DAPO.

In all the five tasks, MAAC+DMAC outperforms MAAC significantly, but MAAC+DMAC does not change the network architecture of MAAC, which shows the benefits of divergence regularizer. As for the result of COMA and COMA+DMAC. We find that COMA+DMAC has higher win rates than COMA in most cases at the end of the training, which can be attributed to the benefits of off-policy learning and exploration of divergence regularizer. Though in some cases COMA learns faster than COMA+DMAC, it falls into sub-optimal in the end. This phenomenon can be observed more clearly in the plots of episode rewards in the hard tasks like 3s5z. This can be an evidence for the advantage of divergence regularizer which helps the agents find a better policy.

The stable policy improvement of divergence regularizer can be manifested by the small variance of the learning curves especially in the comparison between QMIX and QMIX+DMAC. In most tasks, we find that QMIX+DMAC learns substantially faster than QMIX and gets higher win rates in harder tasks. The results of DOP groups are illustrated in the fourth column of Figure 3. DOP+DMAC learns faster than DOP in most cases and finally obtains a better performance. The difference of DOP and DOP+DMAC can also partly show the advantage of naturally off-policy method to the off-policy correction method, as DOP+DMAC replaces the tree backup loss with off-policy TD(λ).

DMAC improves the performance and/or convergence speed of the evaluated algorithms in most tasks. This empirically demonstrates the benefits of divergence regularizer. More-

over, the superiority of our naturally off-policy learning over the biased off-policy correction method can be partly witnessed from the empirical results.

5.2. Comparison with Entropy Regularization

FOP (Zhang et al., 2021) combines value decomposition with entropy regularization, which obtained the state-of-the-art performance in SMAC tasks. FOP has a well-tuned scheme for the temperature parameter of the entropy, so we take FOP as a strong baseline for entropy-regularized methods in cooperative MARL. We compare FOP and FOP+DMAC in five SMAC tasks, 3s_vs_3z, 2s3z, 3s5z, 2c_vs_64zg and MMM2, which respectively correspond to the three levels of difficulties (*i.e.*, easy, hard, and super hard) for SMAC tasks. Three of these tasks are taken from the original paper of FOP. The modifications of FOP+DMAC are also included in Appendix C. The win rates of FOP and FOP+DMAC are illustrated in Figure 4. We can find that FOP+DMAC learns much faster than FOP in 3s_vs_3z, while it performs better than FOP in other four tasks. These results can be an evidence for the advantages of DMAC which could guarantee the monotonic improvement in the original MDP.

6. Conclusion

We propose a multi-agent actor-critic framework, DMAC, for cooperative MARL. We investigate divergence regularization, derive divergence policy iteration, and deduce the update rules for the critic, policy, and target policy in multi-agent settings. DMAC is a naturally off-policy framework and the divergence regularizer is beneficial to exploration and stable policy improvement. DMAC is also a flexible framework and can be combined with many existing MARL algorithms. It is empirically demonstrated that combining DMAC with existing MARL algorithms can improve the performance and convergence speed in a stochastic game and SMAC tasks.

Acknowledgements

References

Agarwal, A., Kumar, S., Sycara, K., and Lewis, M. Learning Transferable Cooperative Behavior in Multi-Agent Team. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2020.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., et al. Impala: Scalable Distributed Deep-Rl with Importance Weighted Actor-Learner Architectures. In *International Conference on Machine Learning (ICML)*, 2018.

Eysenbach, B. and Levine, S. If MaxEnt RL Is The Answer, What Is The Question? *arXiv preprint arXiv:1910.01913*, 2019.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual Multi-Agent Policy Gradients. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Grau-Moya, J., Leibfried, F., and Vrancx, P. Soft Q-Learning with Mutual-Information Regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement Learning with Deep Energy-Based Policies. In *International Conference on Machine Learning (ICML)*, 2017.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with A Stochastic Actor. In *International Conference on Machine Learning (ICML)*, 2018.

Iqbal, S. and Sha, F. Actor-Attention-Critic for Multi-Agent Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, 2019.

Jiang, J., Dun, C., Huang, T., and Lu, Z. Graph Convolutional Reinforcement Learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Kakade, S. and Langford, J. Approximately Optimal Approximate Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, 2002.

Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Abbeel, O. P., and Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. Bridging The Gap Between Value and Policy Based Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. Trust-Pcl: An Off-Policy Trust Region Method for Continuous Control. In *International Conference on Learning Representations (ICLR)*, 2018.

Neu, G., Jonsson, A., and Gómez, V. A Unified View of Entropy-Regularized Markov Decision Processes. *arXiv preprint arXiv:1705.07798*, 2017.

Oliehoek, F. A., Amato, C., et al. *A Concise Introduction To Decentralized POMDPs*, volume 1. Springer, 2016.

Pan, L., Rashid, T., Peng, B., Huang, L., and Whiteson, S. Regularized Softmax Deep Multi-Agent Q-Learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Peng, B., Rashid, T., de Witt, C. A. S., Kamienny, P.-A., Torr, P. H., Böhmer, W., and Whiteson, S. FACMAC: Factored Multi-Agent Centralised Policy Gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Rashid, T., Samvelyan, M., De Witt, C. S., Farquhar, G., Foerster, J., and Whiteson, S. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, 2018.

Samvelyan, M., Rashid, T., de Witt, C. S., Farquhar, G., Nardelli, N., Rudner, T. G., Hung, C.-M., Torr, P. H., Foerster, J. N., and Whiteson, S. The StarCraft Multi-Agent Challenge. In *AAMAS*, 2019.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust Region Policy Optimization. In *International Conference on Machine Learning (ICML)*, 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Son, K., Kim, D., Kang, W. J., Hostallero, D. E., and Yi, Y. QTRAN: Learning To Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, 2019.

Sun, W.-F., Lee, C.-K., and Lee, C.-Y. DFAC Framework: Factorizing The Value Function Via Quantile Mixture for Multi-Agent Distributional Q-Learning. In *International Conference on Machine Learning (ICML)*, 2021.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V. F., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., et al. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2018.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT press, 2018.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2000.

Touati, A., Zhang, A., Pineau, J., and Vincent, P. Stable Policy Optimization Via Off-Policy Divergence Regularization. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2020.

Wang, J., Ren, Z., Liu, T., Yu, Y., and Zhang, C. QPLEX: Duplex Dueling Multi-Agent Q-Learning. In *International Conference on Learning Representations (ICLR)*, 2021a.

Wang, Q., Li, Y., Xiong, J., and Zhang, T. Divergence-Augmented Policy Optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Wang, Y., Han, B., Wang, T., Dong, H., and Zhang, C. DOP: Off-Policy Multi-Agent Decomposed Policy Gradients. In *International Conference on Learning Representations (ICLR)*, 2021b.

Yang, W., Li, X., and Zhang, Z. A Regularized Approach To Sparse Optimal Policy in Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Yang, Y., Hao, J., Liao, B., Shao, K., Chen, G., Liu, W., and Tang, H. Qatten: A General Framework for Cooperative Multiagent Reinforcement Learning. *arXiv preprint arXiv:2002.03939*, 2020.

Yu, C., Velu, A., Vinitsky, E., Wang, Y., Bayen, A. M., and Wu, Y. The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games. *arXiv preprint arXiv:2103.01955*, 2021.

Zhang, T., Li, Y., Wang, C., Xie, G., and Lu, Z. FOP: Factorizing Optimal Joint Policy of Maximum-Entropy Multi-Agent Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, 2021.

A. Proofs

A.1. Proposition 2

Proof. For the sake of simplicity, we define $\tilde{Q}^k = Q^*_{\boldsymbol{\pi}^{k-1}}$ and $\tilde{V}^k = V^*_{\boldsymbol{\pi}^{k-1}}$. From the definition of iteration (3), we have $\boldsymbol{\pi}^k = \boldsymbol{\pi}^*_{\boldsymbol{\pi}^{k-1}}$. Then from Proposition 1, we also have $\tilde{Q}^k \to \tilde{Q}^*$ and $\tilde{V}^k \to \tilde{V}^*$.

From Proposition 5, we have $\boldsymbol{\pi}^k(\boldsymbol{a}|s) \propto \boldsymbol{\pi}^{k-1}(\boldsymbol{a}|s) \exp(\frac{\tilde{Q}^k(s,\boldsymbol{a})}{\omega})$. We define $f^k(s,\boldsymbol{a}) = \exp(\frac{\tilde{Q}^k(s,\boldsymbol{a})}{\omega})$ and $f^*(s,\boldsymbol{a}) = \exp(\frac{\tilde{Q}^*(s,\boldsymbol{a})}{\omega})$, then we have $f^k(s,\boldsymbol{a}) \to f^*(s,\boldsymbol{a})$. Next we will show that $\boldsymbol{\pi}^k(\boldsymbol{a}|s) \propto \boldsymbol{\pi}^0(\boldsymbol{a}|s)\Pi^k_{i=1} f^i(s,\boldsymbol{a})$ or $\boldsymbol{\pi}^k(\boldsymbol{a}|s) = \frac{\boldsymbol{\pi}^0(\boldsymbol{a}|s)\Pi^k_{i=1} f^i(s,\boldsymbol{a})}{Z^k(s)}$ by induction, where $Z^k(s) = \sum_{\boldsymbol{a}'} \boldsymbol{\pi}^0(\boldsymbol{a}'|s)\Pi^k_{i=1} f^i(s,\boldsymbol{a}')$.

$$
\begin{aligned}
\boldsymbol{\pi}^k(\boldsymbol{a}|s) &= \frac{\boldsymbol{\pi}^{k-1}(\boldsymbol{a}|s) f^k(s,\boldsymbol{a})}{\sum_{\boldsymbol{a}'} \boldsymbol{\pi}^{k-1}(\boldsymbol{a}'|s) f^k(s,\boldsymbol{a}')} \\
&= \frac{\frac{\boldsymbol{\pi}^0(\boldsymbol{a}|s)\Pi^{k-1}_{i=1} f^i(s,\boldsymbol{a})}{Z^{k-1}(s)} f^k(s,\boldsymbol{a})}{\sum_{\boldsymbol{a}'} \frac{\boldsymbol{\pi}^0(\boldsymbol{a}'|s)\Pi^{k-1}_{i=1} f^i(s,\boldsymbol{a}')}{Z^{k-1}(s)} f^k(s,\boldsymbol{a}')} \quad \text{(By induction)} \\
&= \frac{\boldsymbol{\pi}^0(\boldsymbol{a}|s)\Pi^k_{i=1} f^i(s,\boldsymbol{a})}{\sum_{\boldsymbol{a}'} \boldsymbol{\pi}^0(\boldsymbol{a}'|s)\Pi^k_{i=1} f^i(s,\boldsymbol{a}')} \\
&= \frac{\boldsymbol{\pi}^0(\boldsymbol{a}|s)\Pi^k_{i=1} f^i(s,\boldsymbol{a})}{Z^k(s)}
\end{aligned}
$$

We define $f^*_s = \max_{\boldsymbol{a}} f^*(s,\boldsymbol{a})$, then we will consider $\tilde{\pi}^*(\boldsymbol{a}|s) = \lim_{k\to\infty} \pi^k(\boldsymbol{a}|s)$.

$$
\begin{aligned}
\lim_{k\to\infty} \pi^k(\boldsymbol{a}|s) &= \lim_{k\to\infty} \frac{\boldsymbol{\pi}^0(\boldsymbol{a}|s)\Pi^k_{i=1} f^i(s,\boldsymbol{a})}{\sum_{\boldsymbol{a}'} \boldsymbol{\pi}^0(\boldsymbol{a}'|s)\Pi^k_{i=1} f^i(s,\boldsymbol{a}')} \\
&= \lim_{k\to\infty} \frac{\boldsymbol{\pi}^0(\boldsymbol{a}|s)\frac{\Pi^k_{i=1} f^i(s,\boldsymbol{a})}{f^*(s,\boldsymbol{a})^k}\frac{f^*(s,\boldsymbol{a})^k}{(f^*_s)^k}}{\sum_{\boldsymbol{a}'} \boldsymbol{\pi}^0(\boldsymbol{a}'|s)\frac{\Pi^k_{i=1} f^i(s,\boldsymbol{a}')}{f^*(s,\boldsymbol{a}')^k}\frac{f^*(s,\boldsymbol{a}')^k}{(f^*_s)^k}} \\
&= \frac{\boldsymbol{\pi}^0(\boldsymbol{a}|s)\lim_{k\to\infty}\frac{\Pi^k_{i=1} f^i(s,\boldsymbol{a})}{f^*(s,\boldsymbol{a})^k}\frac{f^*(s,\boldsymbol{a})^k}{(f^*_s)^k}}{\sum_{\boldsymbol{a}'} \boldsymbol{\pi}^0(\boldsymbol{a}'|s)\lim_{k\to\infty}\frac{\Pi^k_{i=1} f^i(s,\boldsymbol{a}')}{f^*(s,\boldsymbol{a}')^k}\frac{f^*(s,\boldsymbol{a}')^k}{(f^*_s)^k}} \\
&= \mathbb{1}(\boldsymbol{a} \in U_s)\frac{\boldsymbol{\pi}^0(\boldsymbol{a}|s)}{\sum_{\boldsymbol{a}'\in U_s} \boldsymbol{\pi}^0(\boldsymbol{a}'|s)}
\end{aligned}
$$

The last equation is actually from two simple conclusion: (1) If a sequence $\{a_k\} > 0$ and $a_k \to A > 0$, then $\lim_{n\to\infty} \frac{\Pi^n_{k=1} a_k}{A^n} = 1$; (2) For $A > B > 0$, $\lim_{n\to\infty} \frac{B^n}{A^n} = 0$. So $\lim_{k\to\infty} \frac{\Pi^k_{i=1} f^i(s,\boldsymbol{a})}{f^*(s,\boldsymbol{a})^k}\frac{f^*(s,\boldsymbol{a})^k}{(f^*_s)^k} = \mathbb{1}(\boldsymbol{a} \in U_s)$. $\square$

A.2. Theorem 1

We define two operator Γ and Γ_ω as

$$
\Gamma V(s) = \max_{\boldsymbol{\pi}} \sum_{\boldsymbol{a}} \boldsymbol{\pi}(\boldsymbol{a}|s)(r(s,\boldsymbol{a}) + \gamma\mathbb{E}[V(s')])
$$

$$
\Gamma_\omega V(s) = \max_{\boldsymbol{\pi}} \sum_{\boldsymbol{a}} \boldsymbol{\pi}(\boldsymbol{a}|s)(r(s,\boldsymbol{a}) + \gamma\mathbb{E}[V(s')]) - \omega D_{\mathrm{KL}}(\boldsymbol{\pi}(\cdot|s)||\tilde{\boldsymbol{\pi}}^*(\cdot|s))
$$

From the result of traditional Q-learning (Sutton & Barto, 2018), we know that Γ is a γ-contraction and the unique fixed point is the global optimal V-function V^*.

As for Γ_ω, we have the following lemma from Yang et al. (2019):

Lemma 4.

(1) Γ_ω *is a γ-contraction.*

(2) *If $V_1(s) \leq V_2(s)$ for all state s, then $\Gamma_\omega V_1(s) \leq \Gamma_\omega V_2(s)$ for all state s*

(3) *For any constant c, we define $(V + c)(s) = V(s) + c$, then $\Gamma_\omega(V + c)(s) = \Gamma_\omega V(s) + \gamma c$*

Moreover, from the definition we know that $\tilde{V}^k(s) = \max_{\boldsymbol{\pi}} \sum_{\boldsymbol{a}} \boldsymbol{\pi}(\boldsymbol{a}|s)(r(s, \boldsymbol{a}) + \gamma \mathbb{E}[\tilde{V}^k(s')]) - \omega D_{\mathrm{KL}}(\boldsymbol{\pi}(\cdot|s)||\boldsymbol{\pi}^{k-1}(\cdot|s))$ and $\tilde{V}^k \to \tilde{V}^*, \boldsymbol{\pi}^k \to \tilde{\boldsymbol{\pi}}^*$, then we have:

$$\tilde{V}^*(s) = \max_{\boldsymbol{\pi}} \sum_{\boldsymbol{a}} \boldsymbol{\pi}(\boldsymbol{a}|s)(r(s, \boldsymbol{a}) + \gamma \mathbb{E}[\tilde{V}^*(s')]) - \omega D_{\mathrm{KL}}(\boldsymbol{\pi}(\cdot|s)||\tilde{\boldsymbol{\pi}}^*(\cdot|s))$$

This means that $\tilde{V}^*$ is the unique fix point of Γ_ω. Now we have all the tools for the proof of proposition 1. And this proof is inspired by Yang et al. (2019).

Proof.

Given that the initial policy is a uniform distribution, we know from Proposition 2 that:

$$\tilde{\boldsymbol{\pi}}^*(\boldsymbol{a}|s) = \mathbb{1}(\boldsymbol{a} \in U_s)\frac{1}{|U_s|}, \tag{6}$$

where $|U_s|$ is the number of actions contained in the set U_s. Then we consider $D_{\mathrm{KL}}(\boldsymbol{\pi}(\cdot|s)||\tilde{\boldsymbol{\pi}}^*(\cdot|s))$ and have:

$$D_{\mathrm{KL}}(\boldsymbol{\pi}(\cdot|s)||\tilde{\boldsymbol{\pi}}^*(\cdot|s)) = \sum_{\boldsymbol{a} \in U_s} \boldsymbol{\pi}(\boldsymbol{a}|s) \log \frac{\boldsymbol{\pi}(\boldsymbol{a}|s)}{\tilde{\boldsymbol{\pi}}^*(\boldsymbol{a}|s)} \tag{7}$$

$$= \sum_{\boldsymbol{a} \in U_s} \boldsymbol{\pi}(\boldsymbol{a}|s) \log \boldsymbol{\pi}(\boldsymbol{a}|s) - \sum_{\boldsymbol{a} \in U_s} \boldsymbol{\pi}(\boldsymbol{a}|s) \log \tilde{\boldsymbol{\pi}}^*(\boldsymbol{a}|s) \tag{8}$$

$$\leq -\sum_{\boldsymbol{a} \in U_s} \boldsymbol{\pi}(\boldsymbol{a}|s) \log \tilde{\boldsymbol{\pi}}^*(\boldsymbol{a}|s) \quad (\boldsymbol{\pi}(\boldsymbol{a}|s) \leq 1) \tag{9}$$

$$= \log |U_s| \sum_{\boldsymbol{a} \in U_s} \boldsymbol{\pi}(\boldsymbol{a}|s) \tag{10}$$

$$\leq \log |U_s| \quad (\sum_{\boldsymbol{a} \in U_s} \boldsymbol{\pi}(\boldsymbol{a}|s) \leq \sum_{\boldsymbol{a} \in A} \boldsymbol{\pi}(\boldsymbol{a}|s) = 1) \tag{11}$$

$$\leq \log |A| \tag{12}$$

Next we will consider the relation between Γ_ω and Γ:

$$\Gamma_\omega V(s) = \max_{\boldsymbol{\pi}} \sum_{\boldsymbol{a}} \boldsymbol{\pi}(\boldsymbol{a}|s)(r(s, \boldsymbol{a}) + \gamma \mathbb{E}[V(s')]) - \omega D_{\mathrm{KL}}(\boldsymbol{\pi}(\cdot|s)||\tilde{\boldsymbol{\pi}}^*(\cdot|s))$$

$$\leq \max_{\boldsymbol{\pi}} \sum_{\boldsymbol{a}} \boldsymbol{\pi}(\boldsymbol{a}|s)(r(s, \boldsymbol{a}) + \gamma \mathbb{E}[V(s')]) \tag{13}$$

$$= \Gamma V(s)$$

$$\Gamma_\omega V(s) = \max_{\boldsymbol{\pi}} \sum_{\boldsymbol{a}} \boldsymbol{\pi}(\boldsymbol{a}|s)(r(s, \boldsymbol{a}) + \gamma \mathbb{E}[V(s')]) - \omega D_{\mathrm{KL}}(\boldsymbol{\pi}(\cdot|s)||\tilde{\boldsymbol{\pi}}^*(\cdot|s))$$

$$\geq \max_{\boldsymbol{\pi}} \sum_{\boldsymbol{a}} \boldsymbol{\pi}(\boldsymbol{a}|s)(r(s, \boldsymbol{a}) + \gamma \mathbb{E}[V(s')]) - \max_{\boldsymbol{\pi}} D_{\mathrm{KL}}(\boldsymbol{\pi}(\cdot|s)||\tilde{\boldsymbol{\pi}}^*(\cdot|s)) \tag{14}$$

$$\geq \Gamma V(s) - \omega \log |A| \quad \text{(From the inequality (12))}$$

We will show that $\Gamma^k V(s) - \omega \log |A| \sum_{t=0}^{k-1} \gamma^t \leq \Gamma_\omega^k V(s) \leq \Gamma^k V(s)$ for any V and state s by induction.

LHS:

$$\Gamma_\omega^{k+1} V(s) = \Gamma_\omega(\Gamma_\omega^k V(s))$$

$$\geq \Gamma_\omega(\Gamma^k V(s) - \omega \log |A| \sum_{t=0}^{k-1} \gamma^t) \quad \text{(From the induction and the monotonicity in lemma 4)}$$

$$= \Gamma_\omega(\Gamma^k V(s)) - \omega \log |A| \sum_{t=0}^{k-1} \gamma^{t+1} \quad \text{(From the conclusion about constant in lemma 4)}$$

$$\geq \Gamma^{k+1} V(s) - \omega \log |A| - \omega \log |A| \sum_{t=0}^{k-1} \gamma^{t+1} \quad \text{(From the inequality 14)}$$

$$= \Gamma^{k+1} V(s) - \omega \log |A| \sum_{t=0}^{k} \gamma^t$$

RHS:

$$\Gamma_\omega^{k+1} V(s) = \Gamma_\omega(\Gamma_\omega^k V(s))$$

$$\leq \Gamma_\omega(\Gamma^k V(s)) \quad \text{(From the induction and the monotonicity in lemma 4)}$$

$$\leq \Gamma^{k+1} V(s) \quad \text{(From the inequality 13)}$$

Finally, with all these results above, let $k \to \infty$. As both Γ_ω and Γ are γ-contraction, $\Gamma_\omega^k V \to \tilde{V}^*, \Gamma^k V \to V^*$ for any V. We have:

$$\|\tilde{V}^* - V^*\|_\infty \leq \omega \log |A| \sum_{t=0}^{\infty} \gamma^t = \frac{\omega}{1-\gamma} \log |A|$$

$\square$

A.3. Lemma 1

Proof. We define a new reward function

$$r_\rho^{\boldsymbol{\pi}}(s, \boldsymbol{a}) = r(s, \boldsymbol{a}) - \omega \mathbb{E}_{s' \sim P(\cdot|s,\boldsymbol{a})} \left[D_{\mathrm{KL}} \left(\boldsymbol{\pi}(\cdot|s') \| \boldsymbol{\rho}(\cdot|s') \right) \right],$$

then we can rewrite the definition of operator $\Gamma_\rho^{\boldsymbol{\pi}}$ as

$$\Gamma_\rho^{\boldsymbol{\pi}} Q(s, \boldsymbol{a}) = r_\rho^{\boldsymbol{\pi}}(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,\boldsymbol{a}), \boldsymbol{a}' \sim \boldsymbol{\pi}(\cdot|s')} \left[Q(s', \boldsymbol{a}') \right].$$

With this formula, we can apply the traditional convergence result of policy evaluation in Sutton & Barto (2018). $\square$

A.4. Lemma 2

For the proof of Lemma 2, we need the following lemma (Haarnoja et al., 2018) about improving policy in entropy-regularized MDP.

Lemma 5. *If we have a new policy $\boldsymbol{\pi}_{\mathrm{new}}$ and*

$$\boldsymbol{\pi}_{\mathrm{new}} = \arg\min_{\boldsymbol{\pi}} D_{\mathrm{KL}} \left(\boldsymbol{\pi}(\cdot|s) \| \frac{\exp\left(Q_{\mathrm{ent}}^{\boldsymbol{\pi}_{\mathrm{old}}}(s, \cdot)/\omega\right)}{Z^{\boldsymbol{\pi}_{\mathrm{old}}}(s)} \right),$$

where $Z^{\boldsymbol{\pi}_{\mathrm{old}}}(s)$ represents the normalization term, then we have

$$Q_{\mathrm{ent}}^{\boldsymbol{\pi}_{\mathrm{new}}}(s, a) \geq Q_{\mathrm{ent}}^{\boldsymbol{\pi}_{\mathrm{old}}}(s, a), \quad \forall s \in S, a \in A.$$

With Lemma 5, we have the following proof of Lemma 2.

Proof. Let $\hat{Q}$ be the same as the definition in Proof A.7. Then we have $\hat{Q}^{\boldsymbol{\pi}}(s, \boldsymbol{a}) = Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}}(s, \boldsymbol{a}) + \omega \log \boldsymbol{\rho}(\boldsymbol{a}|s)$, $\forall \boldsymbol{\pi}$.

According to Lemma 5,

$$\hat{\boldsymbol{\pi}}_{\text{new}}(s, \cdot) = \arg \min_{\boldsymbol{\pi}} D_{\text{KL}} \left(\boldsymbol{\pi}(\cdot|s) \| \frac{\exp \left(\hat{Q}^{\boldsymbol{\pi}_{\text{old}}}(s, \cdot)/\omega \right)}{Z^{\boldsymbol{\pi}_{\text{old}}}(s)} \right)$$

$$\hat{Q}^{\hat{\boldsymbol{\pi}}_{\text{new}}}(s, \boldsymbol{a}) \geq \hat{Q}^{\boldsymbol{\pi}_{\text{old}}}(s, \boldsymbol{a}), \quad \forall \boldsymbol{a} \in A.$$

With the definition, we have

$$D_{\text{KL}} \left(\boldsymbol{\pi}(\cdot|s) \| \frac{\exp(\hat{Q}^{\boldsymbol{\pi}_{\text{old}}}(s, \cdot)/\omega)}{Z^{\boldsymbol{\pi}_{\text{old}}}(s)} \right) = D_{\text{KL}} \left(\boldsymbol{\pi}(\cdot|s) \| \boldsymbol{\rho}(\cdot|s) \frac{\exp(Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s, \cdot)/\omega)}{Z^{\boldsymbol{\pi}_{\text{old}}}(s)} \right)$$

$$\boldsymbol{\pi}_{\text{new}} = \hat{\boldsymbol{\pi}}_{\text{new}}$$

$$Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{new}}}(s, \boldsymbol{a}) = \hat{Q}^{\boldsymbol{\pi}_{\text{new}}}(s, \boldsymbol{a}) - \omega \log \boldsymbol{\rho}(\boldsymbol{a}|s) \geq \hat{Q}^{\boldsymbol{\pi}_{\text{old}}}(s, \boldsymbol{a}) - \omega \log \boldsymbol{\rho}(\boldsymbol{a}|s) = Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s, \boldsymbol{a}).$$

$\square$

A.5. Lemma 3

Proof. From the equation

$$\pi_{\text{new}}^i = \arg \max_{\pi_i} \sum_{a_i} \pi_i(a_i|s) \left(k_i(s) Q_{\rho}^{\pi_{\text{old}}^i}(s, a_i) - \omega \log \frac{\pi_i(a_i|s)}{\rho_i(a_i|s)} \right),$$

we can obtain

$$\sum_{a_i} \pi_{\text{new}}^i(a_i|s) \left(k_i(s) Q_{\rho}^{\pi_{\text{old}}^i}(s, a_i) - \omega \log \frac{\pi_{\text{new}}^i(a_i|s)}{\rho_i(a_i|s)} \right)$$

$$\geq \sum_{a_i} \pi_{\text{old}}^i(a_i|s) \left(k_i(s) Q_{\rho}^{\pi_{\text{old}}^i}(s, a_i) - \omega \log \frac{\pi_{\text{old}}^i(a_i|s)}{\rho_i(a_i|s)} \right). \tag{15}$$

By taking expectation on the both side of (15), we can obtain the followings.

$$\sum_{a_{-i}} \tilde{\pi}_{-i}(a_{-i}|s) \sum_{a_i} \pi_{\text{new}}^i(a_i|s) \left(k_i(s) Q_{\rho}^{\pi_{\text{old}}^i}(s, a_i) - \omega \log \frac{\pi_{\text{new}}^i(a_i|s)}{\rho_i(a_i|s)} \right)$$

$$\geq \sum_{a_{-i}} \tilde{\pi}_{-i}(a_{-i}|s) \sum_{a_i} \pi_{\text{old}}^i(a_i|s) \left(k_i(s) Q_{\rho}^{\pi_{\text{old}}^i}(s, a_i) - \omega \log \frac{\pi_{\text{old}}^i(a_i|s)}{\rho_i(a_i|s)} \right) \tag{16}$$

$$\forall i \in I \quad \forall \tilde{\pi}_{-i}$$

Moreover, we can easily have the following derivation,

$$\sum_{a_i} \pi_{\text{new}}^i(a_i|s) \left(k_i(s) Q_{\rho}^{\pi_{\text{old}}^i}(s, a_i) + b(s) - \omega \log \frac{\pi_{\text{new}}^i(a_i|s)}{\rho_i(a_i|s)} \right)$$

$$= \sum_{a_{-i}} u_1(a_{-i}|s) \sum_{a_i} \pi_{\text{new}}^i(a_i|s) \left(k_i(s) Q_{\rho}^{\pi_{\text{old}}^i}(s, a_i) + b(s) - \omega \log \frac{\pi_{\text{new}}^i(a_i|s)}{\rho_i(a_i|s)} \right)$$

$$= \sum_{a_{-i}} u_2(a_{-i}|s) \sum_{a_i} \pi_{\text{new}}^i(a_i|s) \left(k_i(s) Q_{\rho}^{\pi_{\text{old}}^i}(s, a_i) + b(s) - \omega \log \frac{\pi_{\text{new}}^i(a_i|s)}{\rho_i(a_i|s)} \right) \tag{17}$$

$$\forall u_1, u_2.$$

Then we have

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{a} \sim \boldsymbol{\pi}_{\text{new}}} \left[Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s, \boldsymbol{a}) - \log \frac{\boldsymbol{\pi}_{\text{new}}(\boldsymbol{a}|s)}{\boldsymbol{\rho}(\boldsymbol{a}|s)} \right] \\
&= \sum_{\boldsymbol{a}} \boldsymbol{\pi}_{\text{new}}(\boldsymbol{a}|s) \sum_{i} \left(k_i(s) Q_{\boldsymbol{\rho}}^{\pi_{\text{old}}^i}(s, a_i) + b(s) - \omega \log \frac{\pi_{\text{new}}^i(a_i|s)}{\rho_i(a_i|s)} \right) \\
&= \sum_{\boldsymbol{a}} \boldsymbol{\pi}_{\text{new}}(\boldsymbol{a}|s) \sum_{i} \left(k_i(s) Q_{\boldsymbol{\rho}}^{\pi_{\text{old}}^i}(s, a_i) + b(s) - \omega \log \frac{\pi_{\text{new}}^i(a_i|s)}{\rho_i(a_i|s)} \right) \\
&= \sum_{i} \sum_{a_{-i}} \pi_{\text{new}}^{-i}(a_{-i}|s) \sum_{a_i} \pi_{\text{new}}^i(a_i|s) \left(k_i(s) Q_{\boldsymbol{\rho}}^{\pi_{\text{old}}^i}(s, a_i) + b(s) - \omega \log \frac{\pi_{\text{new}}^i(a_i|s)}{\rho_i(a_i|s)} \right) \qquad (18) \\
&= \sum_{i} \sum_{a_{-i}} \pi_{\text{old}}^{-i}(a_{-i}|s) \sum_{a_i} \pi_{\text{new}}^i(a_i|s) \left(k_i(s) Q_{\boldsymbol{\rho}}^{\pi_{\text{old}}^i}(s, a_i) + b(s) - \omega \log \frac{\pi_{\text{new}}^i(a_i|s)}{\rho_i(a_i|s)} \right) \\
&\geq \sum_{i} \sum_{a_{-i}} \pi_{\text{old}}^{-i}(a_{-i}|s) \sum_{a_i} \pi_{\text{old}}^i(a_i|s) \left(k_i(s) Q_{\boldsymbol{\rho}}^{\pi_{\text{old}}^i}(s, a_i) + b(s) - \omega \log \frac{\pi_{\text{old}}^i(a_i|s)}{\rho_i(a_i|s)} \right) \\
&= V_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s)
\end{aligned}
$$

The fourth equation is from (17) and the fifth inequality is from (16).

By repeatedly applying (18) and the relation $Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s, \boldsymbol{a}) = r(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s'} \left[V_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s') \right]$, we can complete the proof as followings.

$$
\begin{aligned}
Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s, \boldsymbol{a}) &= r(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s'} \left[V_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s') \right] \\
&\leq r(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s'} \left[\mathbb{E}_{\boldsymbol{a}' \sim \boldsymbol{\pi}_{\text{new}}} \left[Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s', , \boldsymbol{a}') - \log \frac{\boldsymbol{\pi}_{\text{new}}(\boldsymbol{a}'|s')}{\boldsymbol{\rho}(\boldsymbol{a}'|s')} \right] \right] \\
&= r(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s'} \left[\mathbb{E}_{\boldsymbol{a}' \sim \boldsymbol{\pi}_{\text{new}}} \left[r(s', \boldsymbol{a}') + \gamma \mathbb{E}_{s''} \left[V_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s'') \right] - \log \frac{\boldsymbol{\pi}_{\text{new}}(\boldsymbol{a}'|s')}{\boldsymbol{\rho}(\boldsymbol{a}'|s')} \right] \right] \\
&\cdots \\
&\leq Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{new}}}(s, \boldsymbol{a})
\end{aligned}
$$

$\square$

A.6. Theorem 2

Proof. First, we will show that Divergence Policy Iteration will monotonically improve the policy. From Lemma 2, we know that

$$
\begin{aligned}
V_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{new}}}(s) &= \mathbb{E}_{\boldsymbol{a} \sim \boldsymbol{\pi}_{\text{new}}(\cdot|s)} \left[Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{new}}}(s, \boldsymbol{a}) - \omega \log \frac{\boldsymbol{\pi}_{\text{new}}(\boldsymbol{a}|s)}{\boldsymbol{\rho}(\boldsymbol{a}|s)} \right] \\
&\geq \mathbb{E}_{\boldsymbol{a} \sim \boldsymbol{\pi}_{\text{new}}(\cdot|s)} \left[Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s, \boldsymbol{a}) - \omega \log \frac{\boldsymbol{\pi}_{\text{new}}(\boldsymbol{a}|s)}{\boldsymbol{\rho}(\boldsymbol{a}|s)} \right] \\
&\geq \mathbb{E}_{\boldsymbol{a} \sim \boldsymbol{\pi}_{\text{old}}(\cdot|s)} \left[Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s, \boldsymbol{a}) - \omega \log \frac{\boldsymbol{\pi}_{\text{old}}(\boldsymbol{a}|s)}{\boldsymbol{\rho}(\boldsymbol{a}|s)} \right] \\
&= V_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s).
\end{aligned}
$$

The first inequality is from the conclusion of Lemma 2 that

$$
Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{new}}}(s, \boldsymbol{a}) \geq Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s, \boldsymbol{a}) \quad \forall \boldsymbol{a} \in A,
$$

and the second inequality is from the definition of $\boldsymbol{\pi}_{\text{new}}$ that

$$
\boldsymbol{\pi}_{\text{new}} = \arg \min_{\boldsymbol{\pi}} D_{\text{KL}} \left(\boldsymbol{\pi}(\cdot|s) \| \frac{\exp \left(Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s, \cdot)/\omega \right)}{Z^{\boldsymbol{\pi}_{\text{old}}}(s)} \right).
$$

Here we have $V^{\pi_{\text{new}}}(s) \geq V^{\pi_{\text{old}}}(s)$, $\forall s \in S$, and thus $J_{\rho}(\pi_{\text{new}}) \geq J_{\rho}(\pi_{\text{old}})$. So, Divergence Policy Iteration will monotonically improve the policy.

Since the Q_{ρ}^{π} is bounded above (the reward function is bounded), the sequence of Q-function $\{Q^k\}$ of Divergence Policy Iteration will converge and the corresponding policy sequence will also converge to some policy π_{conv}. We need to show $\pi_{\text{conv}} = \pi_{\rho}^*$.

$$V_{\rho}^{\pi_{\text{conv}}}(s) = \mathbb{E}_{\boldsymbol{a} \sim \pi_{\text{conv}}(\cdot|s)} \left[Q_{\rho}^{\pi_{\text{conv}}}(s, \boldsymbol{a}) - \omega \log \frac{\pi_{\text{conv}}(\boldsymbol{a}|s)}{\rho(\boldsymbol{a}|s)} \right]$$

$$\geq \mathbb{E}_{\boldsymbol{a} \sim \pi(\cdot|s)} \left[Q_{\rho}^{\pi_{\text{conv}}}(s, \boldsymbol{a}) - \omega \log \frac{\pi(\boldsymbol{a}|s)}{\rho(\boldsymbol{a}|s)} \right]$$

$$\geq \mathbb{E}_{\boldsymbol{a} \sim \pi(\cdot|s)} \left[r(s, \boldsymbol{a}) + \gamma \mathbb{E}_{\boldsymbol{a}' \sim \pi(\cdot|s')} \left[Q_{\rho}^{\pi_{\text{conv}}}(s', \boldsymbol{a}') - \omega \log \frac{\pi(\boldsymbol{a}'|s')}{\rho(\boldsymbol{a}'|s')} \right] - \omega \log \frac{\pi(\boldsymbol{a}|s)}{\rho(\boldsymbol{a}|s)} \right]$$

$$\cdots$$

$$\geq V_{\rho}^{\pi}(s).$$

The first inequality is from the definition of π_{conv} that

$$\pi_{\text{conv}} = \arg \min_{\pi} D_{\text{KL}} \left(\pi(\cdot|s) \| \frac{\exp \left(Q_{\rho}^{\pi_{\text{conv}}}(s, \cdot)/\omega \right)}{Z^{\pi_{\text{conv}}}(s)} \right)$$

and all the other inequalities are just iteratively using the first inequality and the relation of Q_{ρ}^{π} and V_{ρ}^{π}. With iterations, we replace all the π_{conv} with π in the expression of $V_{\rho}^{\pi_{\text{conv}}}(s)$ and finally we get $V_{\rho}^{\pi}(s)$. Therefore, we have

$$V_{\rho}^{\pi_{\text{conv}}}(s) \geq V_{\rho}^{\pi}(s) \quad \forall s \in S \quad \forall \pi \in \Pi$$
$$J_{\rho}(\pi_{\text{conv}}) \geq J_{\rho}(\pi) \quad \forall \pi \in \Pi$$
$$\pi_{\text{conv}} = \pi_{\rho}^*.$$

$\square$

A.7. Proposition 5

We have the following proposition.

Proposition 5. *If $\pi_{\rho}^* = \arg \max_{\pi} J_{\rho}(\pi)$, and $V_{\rho}^*(s) = V_{\rho}^{\pi_{\rho}^*}(s)$ and $Q_{\rho}^*(s, \boldsymbol{a}) = Q_{\rho}^{\pi_{\rho}^*}(s, \boldsymbol{a})$ are respectively the corresponding Q-function and V-function of π_{ρ}^*, then they satisfy the following properties:*

$$\pi_{\rho}^*(\boldsymbol{a}|s) \propto \rho(\boldsymbol{a}|s) \exp \left(\frac{r(s, \boldsymbol{a}) + \gamma \mathbb{E} \left[V_{\rho}^*(s') \right]}{\omega} \right) \tag{19}$$

$$V_{\rho}^*(s) = \omega \log \sum_{\boldsymbol{a}} \rho(\boldsymbol{a}|s) \exp \left(\frac{r(s, \boldsymbol{a}) + \gamma \mathbb{E} \left[V_{\rho}^*(s') \right]}{\omega} \right) \tag{20}$$

$$Q_{\rho}^*(s, \boldsymbol{a}) = r(s, \boldsymbol{a}) + \gamma \omega \mathbb{E} \left[V_{\rho}^*(s') \right]. \tag{21}$$

Before the proof of Proposition 5, we need some results about the optimal Q-function Q_{ent}^*, the optimal V-function V_{ent}^*, and the optimal policy π_{ent}^* in entropy-regularized MDP. We have the following lemma (Nachum et al., 2017).

Lemma 6.

$$\pi_{\text{ent}}^*(s, a) \propto \exp \left(\left(r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[V_{\text{ent}}^*(s') \right] \right) / \omega \right)$$

$$V_{\text{ent}}^*(s) = \omega \log \sum_{a} \exp \left(\left(r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[V_{\text{ent}}^*(s') \right] \right) / \omega \right)$$

$$Q_{\text{ent}}^*(s, a) = r(s, a) + \gamma \omega \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\log \sum_{a'} \exp \left(Q_{\text{ent}}^*(s', a')/\omega \right) \right]$$

With Lemma 6, we can complete the proof of Proposition 5.

Proof. Let $\hat{r}(s, \boldsymbol{a}) = r(s, \boldsymbol{a}) + \omega \log \boldsymbol{\rho}(\boldsymbol{a}|s)$, we consider the objective function

$$\hat{J}(\boldsymbol{\pi}) = \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t \left(\hat{r}(s_t, \boldsymbol{a}_t) - \omega \log \boldsymbol{\pi}(\boldsymbol{a}_t|s_t) \right) \right].$$

Let $\hat{\boldsymbol{\pi}}^*(\boldsymbol{a}|s), \hat{V}^*(s)$ and $\hat{Q}^*(s, \boldsymbol{a})$ be the corresponding optimal policy, V-function and Q-function of $\hat{J}(\boldsymbol{\pi})$. By definition we can obtain

$$\hat{\boldsymbol{\pi}}^*(\boldsymbol{a}|s) = \boldsymbol{\pi}_{\boldsymbol{\rho}}^*(\boldsymbol{a}|s)$$

$$\hat{V}^*(s) = \mathbb{E}_{\boldsymbol{a} \sim \hat{\boldsymbol{\pi}}^*(\cdot|s), s' \sim P(\cdot|s,a)} \left[\hat{r}(s, \boldsymbol{a}) + \gamma \hat{V}^*(s') - \omega \log \hat{\boldsymbol{\pi}}^*(\boldsymbol{a}|s) \right]$$

$$= \mathbb{E}_{\boldsymbol{a} \sim \boldsymbol{\pi}_{\boldsymbol{\rho}}^*(\cdot|s), s' \sim P(\cdot|s,a)} \left[r(s, \boldsymbol{a}) + \gamma \hat{V}^*(s') - \omega \log \frac{\boldsymbol{\pi}_{\boldsymbol{\rho}}^*(\boldsymbol{a}|s)}{\boldsymbol{\rho}(\boldsymbol{a}|s)} \right]$$

$$= V_{\boldsymbol{\rho}}^*(s)$$

$$\hat{Q}^*(s, \boldsymbol{a}) = \hat{r}(s, \boldsymbol{a}) + \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\hat{V}^*(s') \right]$$

$$= r(s, \boldsymbol{a}) + \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[V_{\boldsymbol{\rho}}^*(s') \right] + \omega \log \boldsymbol{\rho}(\boldsymbol{a}|s)$$

$$= Q_{\boldsymbol{\rho}}^*(s, \boldsymbol{a}) + \omega \log \boldsymbol{\rho}(\boldsymbol{a}|s).$$

According to Lemma 6, we have

$$\boldsymbol{\pi}_{\boldsymbol{\rho}}^*(\boldsymbol{a}|s) = \hat{\boldsymbol{\pi}}^*(\boldsymbol{a}|s) \propto \exp \left(\left(\hat{r}(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\hat{V}^*(s') \right] \right) / \omega \right)$$

$$= \boldsymbol{\rho}(\boldsymbol{a}|s) \exp \left(\left(r(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[V_{\boldsymbol{\rho}}^*(s') \right] \right) / \omega \right)$$

$$V_{\boldsymbol{\rho}}^*(s) = \hat{V}^*(s)$$

$$= \omega \log \sum_{\boldsymbol{a}} \exp \left(\left(\hat{r}(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\hat{V}^*(s') \right] \right) / \omega \right)$$

$$= \omega \log \sum_{\boldsymbol{a}} \boldsymbol{\rho}(\boldsymbol{a}|s) \exp \left(\left(r(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[V_{\boldsymbol{\rho}}^*(s') \right] \right) / \omega \right)$$

$$Q_{\boldsymbol{\rho}}^*(s, \boldsymbol{a}) = \hat{Q}^*(s, \boldsymbol{a}) - \omega \log \boldsymbol{\rho}(\boldsymbol{a}|s)$$

$$= \hat{r}(s, \boldsymbol{a}) + \gamma \omega \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\log \sum_{\boldsymbol{a}'} \exp \left(\hat{Q}^*(s', \boldsymbol{a}')/\omega \right) \right] - \omega \log \boldsymbol{\rho}(\boldsymbol{a}|s)$$

$$= r(s, \boldsymbol{a}) + \gamma \omega \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\log \sum_{\boldsymbol{a}'} \boldsymbol{\rho}(\boldsymbol{a}|s) \exp \left(Q_{\boldsymbol{\rho}}^*(s', \boldsymbol{a}')/\omega \right) \right].$$

$\square$

A.8. Proposition 4

Proof. From Proposition 5, we can obtain

$$\boldsymbol{\pi}_{\boldsymbol{\rho}}^*(\boldsymbol{a}|s) = \frac{\boldsymbol{\rho}(\boldsymbol{a}|s) \exp \left(\left(r(s, \boldsymbol{a}) + \gamma \mathbb{E} \left[V_{\boldsymbol{\rho}}^*(s') \right] \right) / \omega \right)}{\sum_{\boldsymbol{b}} \boldsymbol{\rho}(\boldsymbol{b}|s) \exp \left(\left(r(s, \boldsymbol{b}) + \gamma \mathbb{E} \left[V_{\boldsymbol{\rho}}^*(s') \right] \right) / \omega \right)}$$

$$= \frac{\boldsymbol{\rho}(\boldsymbol{a}|s) \exp \left(Q_{\boldsymbol{\rho}}^*(s, \boldsymbol{a})/\omega \right)}{\exp \left(V_{\boldsymbol{\rho}}^*(s)/\omega \right)}$$

$$= \boldsymbol{\rho}(\boldsymbol{a}|s) \exp \left(\left(Q_{\boldsymbol{\rho}}^*(s, \boldsymbol{a}) - V_{\boldsymbol{\rho}}^*(s) \right) / \omega \right). \tag{22}$$

By rearranging the equation, we have

$$V_{\boldsymbol{\rho}}^*(s) = Q_{\boldsymbol{\rho}}^*(s, \boldsymbol{a}) - \omega \log \frac{\pi_{\boldsymbol{\rho}}^*(\boldsymbol{a}|s)}{\rho(\boldsymbol{a}|s)}, \tag{23}$$

which is tenable for all actions $\boldsymbol{a} \in A$. □

A.9. Derivation of Gradient

$$\nabla_{\theta_i} \mathcal{L}_{\boldsymbol{\pi}} = \mathbb{E}_{s \sim \mathcal{D}} \left[\sum_{\boldsymbol{a}} \nabla_{\theta_i} \boldsymbol{\pi}(\boldsymbol{a}|s) \left(Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s, \boldsymbol{a}) - \omega \log \frac{\boldsymbol{\pi}(\boldsymbol{a}|s)}{\rho(\boldsymbol{a}|s)} \right) + \boldsymbol{\pi}(\boldsymbol{a}|s) \nabla_{\theta_i} (-\omega \log \frac{\boldsymbol{\pi}(\boldsymbol{a}|s)}{\rho(\boldsymbol{a}|s)}) \right]$$

$$= \mathbb{E}_{s \sim \mathcal{D}} \left[\sum_{\boldsymbol{a}} \boldsymbol{\pi}(\boldsymbol{a}|s) \nabla_{\theta_i} \log \pi_i(a_i|s) \left(Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s, \boldsymbol{a}) - \omega \log \frac{\boldsymbol{\pi}(\boldsymbol{a}|s)}{\rho(\boldsymbol{a}|s)} \right) - \omega \boldsymbol{\pi}(\boldsymbol{a}|s) \nabla_{\theta_i} \log \pi_i(a_i|s) \right]$$

$$= \mathbb{E}_{s \sim \mathcal{D}, \boldsymbol{a} \sim \boldsymbol{\pi}} \left[\nabla_{\theta_i} \log \pi_i(a_i|s) \left(Q_{\boldsymbol{\rho}}^{\boldsymbol{\pi}_{\text{old}}}(s, \boldsymbol{a}) - \omega \log \frac{\boldsymbol{\pi}(\boldsymbol{a}|s)}{\rho(\boldsymbol{a}|s)} - \omega \right) \right]$$

A.10. Theoretical Results for the Moving Average Case.

In the case where we take the target policies $\{\rho^t\}$ as the moving average of the policies $\{\pi^t\}$, we will formulate the iterations as followings:

$$\rho^t = (1 - \tau)\rho^{t-1} + \tau\pi^{t-1}$$

$$\pi^t = \arg\max_{\boldsymbol{\pi}} \sum_{s,\boldsymbol{a}} \mu_{\boldsymbol{\pi}}(s, \boldsymbol{a}) r(s, \boldsymbol{a}) - \omega D_C (\mu_{\boldsymbol{\pi}} \| \mu_{\rho^t}) .$$

Then we have:

$$J(\boldsymbol{\pi}^{t+1}) \geq J(\boldsymbol{\pi}^{t+1}) - \omega D_C (\mu_{\boldsymbol{\pi}^{t+1}} \| \mu_{\rho^{t+1}}) = J_{\rho^{t+1}}(\boldsymbol{\pi}^{t+1})$$

$$\geq J(\boldsymbol{\pi}^t) - \omega D_C (\mu_{\boldsymbol{\pi}^t} \| \mu_{\rho^{t+1}})$$

$$\geq J(\boldsymbol{\pi}^t) - (1 - \tau)\omega D_C (\mu_{\boldsymbol{\pi}^t} \| \mu_{\rho^t})$$

$$\geq J(\boldsymbol{\pi}^t) - \omega D_C (\mu_{\boldsymbol{\pi}^t} \| \mu_{\rho^t}) = J_{\rho^t}(\boldsymbol{\pi}^t).$$

The third inequality could be obtained as following:

$$D_C (\mu_{\boldsymbol{\pi}^t} \| \mu_{\rho^{t+1}}) = \sum_{s,\boldsymbol{a}} \mu_{\boldsymbol{\pi}}(s, \boldsymbol{a}) \log \frac{\pi^t(\boldsymbol{a}|s)}{\rho^{t+1}(\boldsymbol{a}|s)}$$

$$= \sum_{s,\boldsymbol{a}} \mu_{\boldsymbol{\pi}}(s, \boldsymbol{a}) \log \frac{\pi^t(\boldsymbol{a}|s)}{\tau\pi^t(\boldsymbol{a}|s) + (1 - \tau)\rho^t(\boldsymbol{a}|s)}$$

$$\leq \sum_{s,\boldsymbol{a}} \mu_{\boldsymbol{\pi}}(s, \boldsymbol{a}) \left[\log \pi^t(\boldsymbol{a}|s) - \left(\tau \log \pi^t(\boldsymbol{a}|s) + (1 - \tau) \log \rho^t(\boldsymbol{a}|s) \right) \right] \qquad \text{from the concavity of } \log(\cdot)$$

$$= (1 - \tau) \sum_{s,\boldsymbol{a}} \mu_{\boldsymbol{\pi}}(s, \boldsymbol{a}) \log \frac{\pi^t(\boldsymbol{a}|s)}{\rho^t(\boldsymbol{a}|s)}$$

$$= (1 - \tau) D_C (\mu_{\boldsymbol{\pi}^t} \| \mu_{\rho^t}) .$$

So the sequence $\{J_{\rho^t}(\boldsymbol{\pi}^t)\}$ is still bounded and improving monotonically and will converge. With this result, we have $\tilde{Q}^k = Q_{\rho^k}^*$ and $\tilde{V}^k = V_{\rho^k}^*$ will converge to $\tilde{Q}^*$ and $\tilde{V}^*$.

Next we will consider the convergence of the sequence $\{\rho^t\}$ and $\{\pi^t\}$. From Proposition 5, we have the formulation that

$$\pi^t(\boldsymbol{a}|s) = \frac{\rho^t(\boldsymbol{a}|s) f^t(s, \boldsymbol{a})}{\sum_{\boldsymbol{b}} \rho^t(\boldsymbol{b}|s) f^t(s, \boldsymbol{b})} \tag{24}$$

where $f^t(s, \boldsymbol{a}) = \exp(\frac{\tilde{Q}^t(s,\boldsymbol{a})}{\omega})$ and $f^t(s, \boldsymbol{a}) \to f^*(s, \boldsymbol{a}) = \exp(\frac{\tilde{Q}^*(s,\boldsymbol{a})}{\omega})$.

We define $Z^t(s) = \sum_{\boldsymbol{b}} \rho^t(\boldsymbol{b}|s) f^t(s, \boldsymbol{b})$ and will show that $\{Z^t(s)\}$ will converge for each state s. Actually, we will show that $\{Z^t(s)\}$ monotonically improves and is bounded.

With $Z^t(s)$, we could rewrite the the relation between ρ^{t+1} and ρ^t as followings:

$$\rho^{t+1}(\boldsymbol{a}|s) = \left(1 - \tau + \tau \frac{f^t(s, \boldsymbol{a})}{Z^t(s)}\right)\rho^t(\boldsymbol{a}|s). \tag{25}$$

Then we have:

$$
\begin{aligned}
Z^{t+1}(s) - Z^t(s) &= \sum_{\boldsymbol{b}} \rho^{t+1}(\boldsymbol{b}|s) f^{t+1}(s, \boldsymbol{b}) - \sum_{\boldsymbol{b}} \rho^t(\boldsymbol{b}|s) f^t(s, \boldsymbol{b}) \\
&\geq \sum_{\boldsymbol{b}} \rho^{t+1}(\boldsymbol{b}|s) f^t(s, \boldsymbol{b}) - - \sum_{\boldsymbol{b}} \rho^t(\boldsymbol{b}|s) f^t(s, \boldsymbol{b}) \quad (f^{t+1}(s, \boldsymbol{b}) \geq f^t(s, \boldsymbol{b}) \text{ from } \{\tilde{Q}^t\} \text{ improve monotonically.}) \\
&= \tau \sum_{\boldsymbol{b}} \frac{f^t(s, \boldsymbol{b}) - Z^t(s)}{Z^t(s)} \rho^t(\boldsymbol{b}|s) f^t(s, \boldsymbol{b}) \qquad \text{from iteration (25)} \\
&= \tau \frac{\sum_{\boldsymbol{b}} \rho^t(\boldsymbol{b}|s) f^t(s, \boldsymbol{b})^2 - Z^t(s)^2}{Z^t(s)} \\
&= \tau \frac{\mathrm{Var}_{\boldsymbol{b} \sim \rho^t(\cdot|s)}[f^t(s, \boldsymbol{b})]}{Z^t(s)} \\
&\geq 0.
\end{aligned}
$$

Moreover, let $M^* = \max_{s,\boldsymbol{a}} f^*(s, \boldsymbol{a})$. When t is sufficiently large we have $f^t(s, \boldsymbol{a}) \leq f^*(s, \boldsymbol{a}) + 1 \leq M^* + 1$. So $Z^t(s) \leq M^* + 1$ when t is sufficiently large. With all the above discussions, we show that $Z^t(s)$ converge to $Z^*(s)$.

Next we will divide the actions into three classes according to the relation between $f^*(s, \boldsymbol{a})$ and $Z^*(s)$, given any fixed state s. Let $I_s^+ = \{\boldsymbol{a} \in A | f^*(s, \boldsymbol{a}) > Z^*(s)\}$, $I_s^- = \{\boldsymbol{a} \in A | f^*(s, \boldsymbol{a}) < Z^*(s)\}$, and $I_s^0 = \{\boldsymbol{a} \in A | f^*(s, \boldsymbol{a}) = Z^*(s)\}$. We will show three properties about these sets:

(1) $\forall \boldsymbol{a} \in I_s^-, \quad \rho^t(\boldsymbol{a}|s) \to 0$.

(2) $I_s^+ = \emptyset$.

(3) $\sum_{\boldsymbol{a} \in I_s^0} \rho^t(\boldsymbol{a}|s) \to 1$.

It is obvious that the property (3) is an corollary of property (1) and (2). So we will focus on property (1) and (2).

As for property (1), consider any action $\boldsymbol{a} \in I_s^-$. Let $\epsilon = \frac{Z^*(s) - f^*(s,\boldsymbol{a})}{4} > 0$, when t is sufficiently large, we have $f^t(s, \boldsymbol{a}) < f^*(s, \boldsymbol{a}) + \epsilon = Z^*(s) - 3\epsilon$ and $Z^t(s) > Z^*(s) - \epsilon$. Then we have

$$
\begin{aligned}
\rho^{t+1}(\boldsymbol{a}|s) &= \left(1 - \tau + \tau \frac{f^t(s, \boldsymbol{a})}{Z^t(s)}\right)\rho^t(\boldsymbol{a}|s) \\
&< \left(1 - \tau + \tau \frac{Z^*(s) - 3\epsilon}{Z^*(s) + \epsilon}\right)\rho^t(\boldsymbol{a}|s) \\
&= \left(1 - \frac{4\epsilon\tau}{Z^*(s) + \epsilon}\right)\rho^t(\boldsymbol{a}|s).
\end{aligned}
$$

Since the constant $1 - \frac{4\epsilon\tau}{Z^*(s)+\epsilon} < 1$, we know that $\rho^t(\boldsymbol{a}|s) \to 0$ for action $\boldsymbol{a}$.

As for property (2), suppose that $I_s^+ \neq \emptyset$. Then, we could take some action $\boldsymbol{a} \in I_s^+$. Let $\epsilon = \frac{f^*(s,\boldsymbol{a}) - Z^*(s)}{4} > 0$, when t is

sufficiently large, we have $f^t(s, \boldsymbol{a}) > f^*(s, \boldsymbol{a}) - \epsilon$ and $Z^t(s) < Z^*(s) + \epsilon = f^*(s, \boldsymbol{a}) - 3\epsilon$. Then we have

$$
\begin{aligned}
\boldsymbol{\rho}^{t+1}(\boldsymbol{a}|s) &= \left(1 - \tau + \tau \frac{f^t(s, \boldsymbol{a})}{Z^t(s)}\right) \boldsymbol{\rho}^t(\boldsymbol{a}|s) \\
&> \left(1 - \tau + \tau \frac{f^*(s, \boldsymbol{a}) - \epsilon}{f^*(s, \boldsymbol{a}) - 3\epsilon}\right) \boldsymbol{\rho}^t(\boldsymbol{a}|s) \\
&= \left(1 + \frac{2\epsilon\tau}{f^*(s, \boldsymbol{a}) - 3\epsilon}\right) \boldsymbol{\rho}^t(\boldsymbol{a}|s).
\end{aligned}
$$

Since the constant $1 + \frac{2\epsilon\tau}{f^*(s,\boldsymbol{a})-3\epsilon} > 1$, we know that $\boldsymbol{\rho}^t(\boldsymbol{a}|s) \to \infty$ for action $\boldsymbol{a}$. But we know that $\boldsymbol{\rho}^t(\boldsymbol{a}|s) \le 1$ which is a contradiction.

From these three properties we actually know that $Z^*(s) = \max_{\boldsymbol{a}} f^*(s, \boldsymbol{a})$. The proof is as followings: from property (2) we know that $f^*(s, \boldsymbol{a}) \le Z^*(s), \forall \boldsymbol{a} \in A$; suppose that $f^*(s, \boldsymbol{a}) < Z^*(s), \forall \boldsymbol{a} \in A$, then from property (1) we know that $\sum_{\boldsymbol{a}} \boldsymbol{\rho}^t(\boldsymbol{a}|s) \to 0$ which contradicts to $\sum_{\boldsymbol{a}} \boldsymbol{\rho}^t(\boldsymbol{a}|s) = 1$. This fact could also be written as $I_s^0 = U_s$, where U_s is the same definition as the proof for Proposition 2 in Appendix A.1

Finally, with all these preparations above, we could discuss the convergence of $\boldsymbol{\pi}^t$.

From iteration (25), we could obtain the following formula:

$$
\boldsymbol{\rho}^t(\boldsymbol{a}|s) = \boldsymbol{\rho}^0(\boldsymbol{a}|s) f^t(s, \boldsymbol{a}) \Pi_{k=0}^t \left(1 - \tau + \tau \frac{f^k(s, \boldsymbol{a})}{Z^k(s)}\right) = \boldsymbol{\pi}^0(\boldsymbol{a}|s) f^t(s, \boldsymbol{a}) \Pi_{k=0}^t \left(1 - \tau + \tau \frac{f^k(s, \boldsymbol{a})}{Z^k(s)}\right). \tag{26}
$$

Combining this with (24), we have

$$
\begin{aligned}
\boldsymbol{\pi}^t(\boldsymbol{a}|s) &= \frac{\boldsymbol{\pi}^0(\boldsymbol{a}|s) f^t(s, \boldsymbol{a}) \Pi_{k=0}^{t-1} \left(1 - \tau + \tau \frac{f^k(s,\boldsymbol{a})}{Z^k(s)}\right)}{\sum_{\boldsymbol{b}} \boldsymbol{\pi}^0(\boldsymbol{b}|s) f^t(s, \boldsymbol{b}) \Pi_{k=0}^{t-1} \left(1 - \tau + \tau \frac{f^k(s,\boldsymbol{b})}{Z^k(s)}\right)} \\
&= \frac{\boldsymbol{\pi}^0(\boldsymbol{a}|s) f^t(s, \boldsymbol{a}) \Pi_{k=0}^{t-1} \left((1 - \tau) Z^k(s) + \tau f^k(s, \boldsymbol{a})\right)}{\sum_{\boldsymbol{b}} \boldsymbol{\pi}^0(\boldsymbol{b}|s) f^t(s, \boldsymbol{b}) \Pi_{k=0}^{t-1} \left((1 - \tau) Z^k(s) + \tau f^k(s, \boldsymbol{b})\right)} \\
&= \frac{\boldsymbol{\pi}^0(\boldsymbol{a}|s) \frac{f^t(s,\boldsymbol{a}) \Pi_{k=0}^{t-1} \left((1-\tau)Z^k(s) + \tau f^k(s,\boldsymbol{a})\right)}{(Z^*(s))^{t+1}}}{\sum_{\boldsymbol{b}} \boldsymbol{\pi}^0(\boldsymbol{b}|s) \frac{f^t(s,\boldsymbol{b}) \Pi_{k=0}^{t-1} \left((1-\tau)Z^k(s) + \tau f^k(s,\boldsymbol{b})\right)}{(Z^*(s))^{t+1}}}.
\end{aligned}
$$

We also have

$$
\lim_{t \to \infty} \left((1 - \tau) Z^t(s) + \tau f^t(s, \boldsymbol{a})\right) = \begin{cases} Z^*(s) & \boldsymbol{a} \in I_s^0 \\ (1 - \tau) Z^*(s) + \tau f^*(s, \boldsymbol{a}) < Z^*(s) & \boldsymbol{a} \in I_s^- \end{cases}. \tag{27}
$$

Similarly to the proof for Proposition 2, we have

$$
\lim_{t \to \infty} \frac{f^t(s, \boldsymbol{a}) \Pi_{k=0}^{t-1} \left((1 - \tau) Z^k(s) + \tau f^k(s, \boldsymbol{a})\right)}{(Z^*(s))^{t+1}} = \begin{cases} 1 & \boldsymbol{a} \in I_s^0 \\ 0 & \boldsymbol{a} \in I_s^- \end{cases}. \tag{28}
$$

Finally we obtain

$$
\tilde{\boldsymbol{\pi}}^*(\boldsymbol{a}|s) = \lim_{t \to \infty} \boldsymbol{\pi}^t(\boldsymbol{a}|s) = \mathbb{1}(\boldsymbol{a} \in I_s^0) \frac{\boldsymbol{\pi}^0(\boldsymbol{a}|s)}{\sum_{\boldsymbol{b} \in I_s^0} \boldsymbol{\pi}^0(\boldsymbol{b}|s)} = \mathbb{1}(\boldsymbol{a} \in U_s) \frac{\boldsymbol{\pi}^0(\boldsymbol{a}|s)}{\sum_{\boldsymbol{b} \in U_s} \boldsymbol{\pi}^0(\boldsymbol{b}|s)}. \tag{29}
$$

This conclusion is actually the same as Proposition 2.

Moreover, as $\boldsymbol{\rho}^t$ is the moving average of $\boldsymbol{\pi}^t$, it is obvious that $\boldsymbol{\rho}^t \to \tilde{\boldsymbol{\pi}}^*$. With this result, we could define the same operator Γ_ω as the proof for Theorem 1 in Appendix A.2 and use the same proof to show that $\tilde{V}^*$ is the fixed point of Γ_ω and finally obtain the same conclusion of Theorem 1.

Algorithm 1 DMAC

1: **for** episode = 1 to m **do**
2:     Initialize the environment and receive initial state s
3:     **for** $t = 1$ to max-episode-length **do**
4:         For each agent i, select action $a_i \sim \pi_i(\cdot|s)$
5:         Execute joint-action $\boldsymbol{a} = (a_1, a_2, \cdots, a_n)$ and observe reward r and next state s'
6:         Store $(s, \boldsymbol{a}, r, s')$ in replay buffer $\mathcal{D}$
7:     **end for**
8:     Sample a random minibatch of K samples from $\mathcal{D}$, $\{(s_k, \boldsymbol{a}_k, r_k, s'_k)\}_K$
9:     **for** agent $i = 1$ to n **do**
10:        Update policy π_i: $\theta_i = \theta_i + \beta \nabla_{\theta_i} \mathcal{L}_{\boldsymbol{\pi}}$
11:        Update target policy ρ_i: $\tilde{\theta}_i = (1 - \tau)\tilde{\theta}_i + \tau\theta_i$
12:     **end for**
13:     Update critic: $\phi = \phi - \alpha \nabla_\phi \mathcal{L}_Q$
14:     Update target critic: $\tilde{\phi} = (1 - \tau)\tilde{\phi} + \tau\phi$
15: **end for**

B. Algorithm

Algorithm 1 gives the training procedure of DMAC.

C. Implementation Details

SMAC is an MARL environment based on the game StarCraft II (SC2). Agents control different units in SC2 and can attack, move or take other actions. The general mode of SMAC tasks is that agents control a team of units to counter another team controlled by built-in AI. The target of agents is to wipe out all the enemy units and agents will gain rewards when hurting or killing enemy units. Agents have an observation range and can only observe information of units in this range, but the information of all the units can be accessed in training. We test all the methods in totally 8 tasks/maps: 3m, 2s3z, 3s5z, 8m, 1c3s5z, 3s_vs_3z, 2c_vs_64zg, and MMM2.

C.1. Modifications of the Baseline Methods

The modifications of the baseline methods, COMA, MAAC, QMIX, DOP, and FOP, are as follows:

- **COMA.** We keep the original critic and actor networks and add a target policy network with the same architecture as the actor. As COMA is on-policy but COMA+DMAC is off-policy, we add a replay buffer for experience replay.

- **MAAC** already has a target policy for stability, so we do not need to modify the network architecture. We only change the update rule for the critic and actors.

- **QMIX** is a value-based method, so we need to add a policy network and a target policy network for each agent. We keep the original individual Q-functions to learn the critic in QMIX+DMAC. In divergence-regularized MDP, the max operator is not needed in the critic update, so we abandon the hypernet and use an MLP, which takes individual Q-values and state as input and produces the joint Q-value. This architecture is simple and its expressive capability is not limited by QMIX's IGM condition.

- **DOP.** We keep the original critic and actor networks and add a target policy network with the same architecture as the actor. We keep the value decomposition structure and use off-policy TD(λ) for all samples in training to replace the tree backup loss and on-policy TD(λ) loss.

- **FOP.** We replace the entropy regularizers with divergence regularizers in FOP and use the update rules of DMAC. We keep the original architecture of FOP.
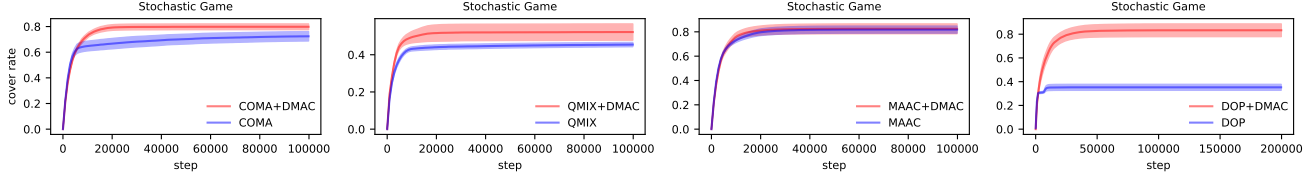
Figure 5. Learning curves in terms of cover rates of COMA, MAAC, QMIX and DOP groups in the randomly generated stochastic game.
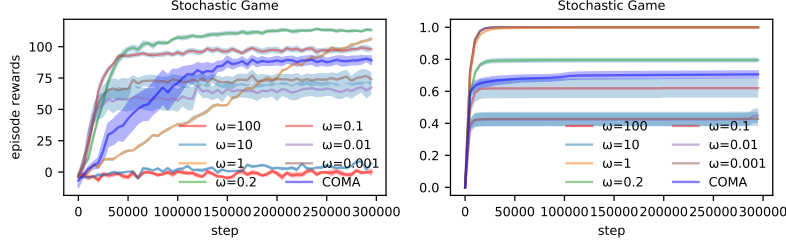

Figure 6. Learning curves in terms of cover rates of COMA and COMA+DMAC with different ω in the randomly generated stochastic game.

C.2. Hyperparameters

As all tasks in our experiments are cooperative with shared reward, so we use parameter-sharing policy network and critic network for MAAC and MAAC+DMAC to accelerate training. Besides, we add a RNN layer to the policy network and critic network in MAAC and MAAC+DMAC to settle the partial observability.

All the policy networks are the same as two linear layers and one GRUCell layer with ReLU activation and the number of hidden units is 64. The individual Q-networks for QMIX group is the same as the policy network mentioned before. The critic network for COMA group is a MLP with three 128-unit hidden layers and ReLU activation. The attention dimension in the critic networks of MAAC group is 32. The number of hidden units of mixer network in QMIX group is 32. The learning rate for critic is 10^{-3} and the learning rate for actor is 10^{-4}. We train all networks with RMSprop optimizer. The discouted factor is $\gamma = 0.99$. The coefficient of regularizer is $\omega = 0.01$ for SMAC tasks and $\omega = 0.2$ for the stochastic game. The *td_lambda* factor used in COMA group is 0.8. The parameter used for soft updating target policy is $\tau = 0.01$. Our code is based on the implementation of PyMARL (Samvelyan et al., 2019), MAAC (Iqbal & Sha, 2019), DOP (Wang et al., 2021b), FOP (Zhang et al., 2021) and an open source code for algorithms in SMAC (https://github.com/starry-sky6688/StarCraft).

C.3. Experiment Settings

We trained each algorithms for five runs with different random seeds. In SMAC tasks, we train each algorithm for one million steps in each run for COMA, QMIX, MAAC, and DOP groups (except 2c_vs_64zg and MMM2) and two million steps for FOP groups. We evaluate 20 episodes in every 10000 training steps in the one million steps training procedure and in every 20000 steps in the two million steps training procedure. In evaluation, we select greedy actions for QMIX and FOP (following the setting in the FOP paper) and sample actions according to action distribution for stochastic policy (COMA, MAAC, DOP and divergence-regularized methods). We do all the experiments by a server with 2 NVIDIA A100 GPUs.

D. Additional Results

Figure 5 shows the learning curves in terms of cover rates of COMA, QMIX, MAAC and DOP groups in the randomly generated stochastic game.

Figure 6 shows learning curves in terms of cover rates and episode rewards of COMA and COMA+DMAC with different ω in the randomly generated stochastic game.

Figure 7 and Figure 8 shows the learning curves of COMA, MAAC, QMIX and DOP groups in terms of mean episode rewards and win rates in seven SMAC maps.

Figure 9 shows the learning curves of FOP+DMAC and FOP in terms of mean episode rewards in 3s_vs_3z,2s3z,3s5z, 2c_vs_64zg and MMM2.
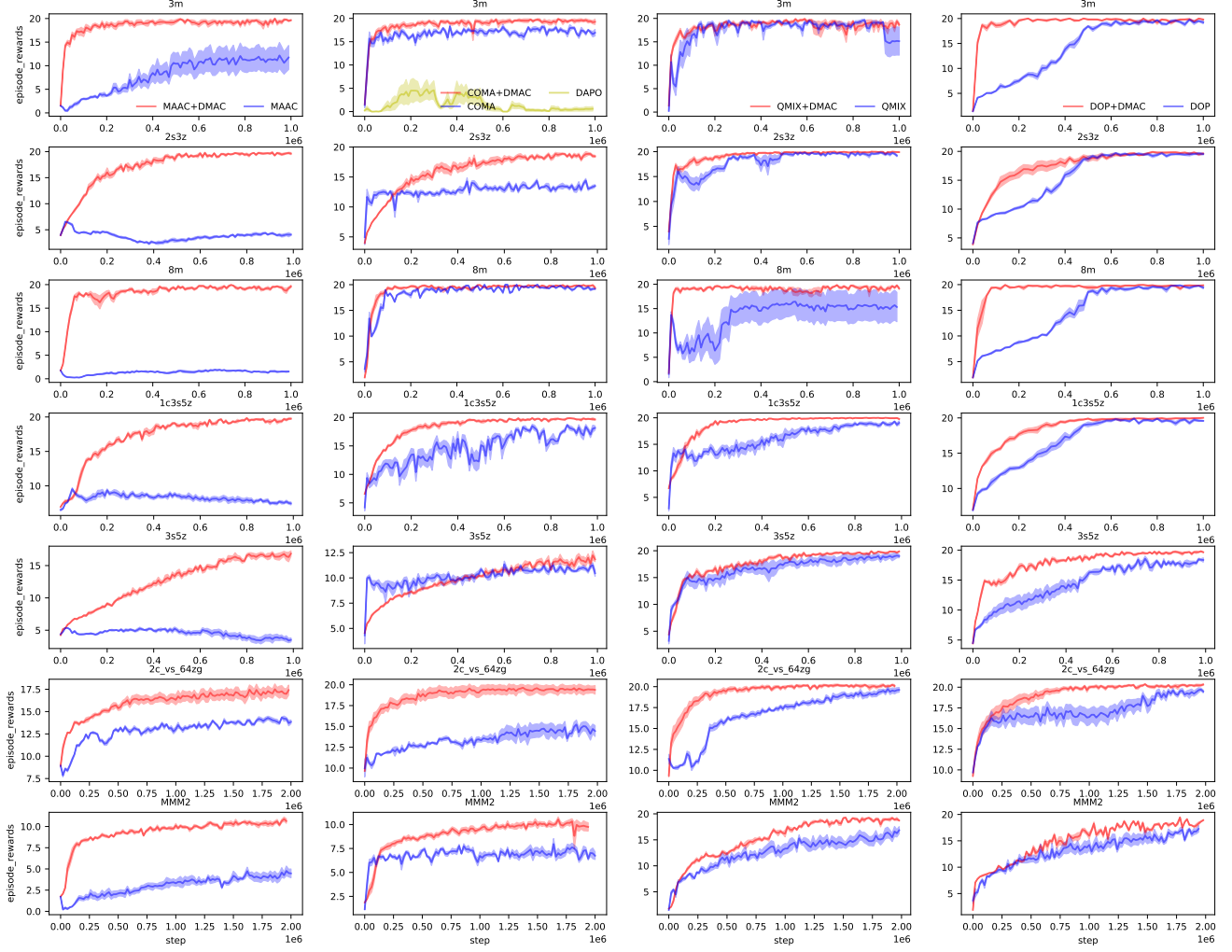
Figure 7. Learning curves in terms of mean episode reward of COMA, MAAC, QMIX, and DOP groups in seven SMAC maps (each row corresponds to a map and each column corresponds to a group).

Figure 10 shows the learning curves in terms of mean episode rewards of COMA and DOP groups in the CDM environment used by the DOP paper.
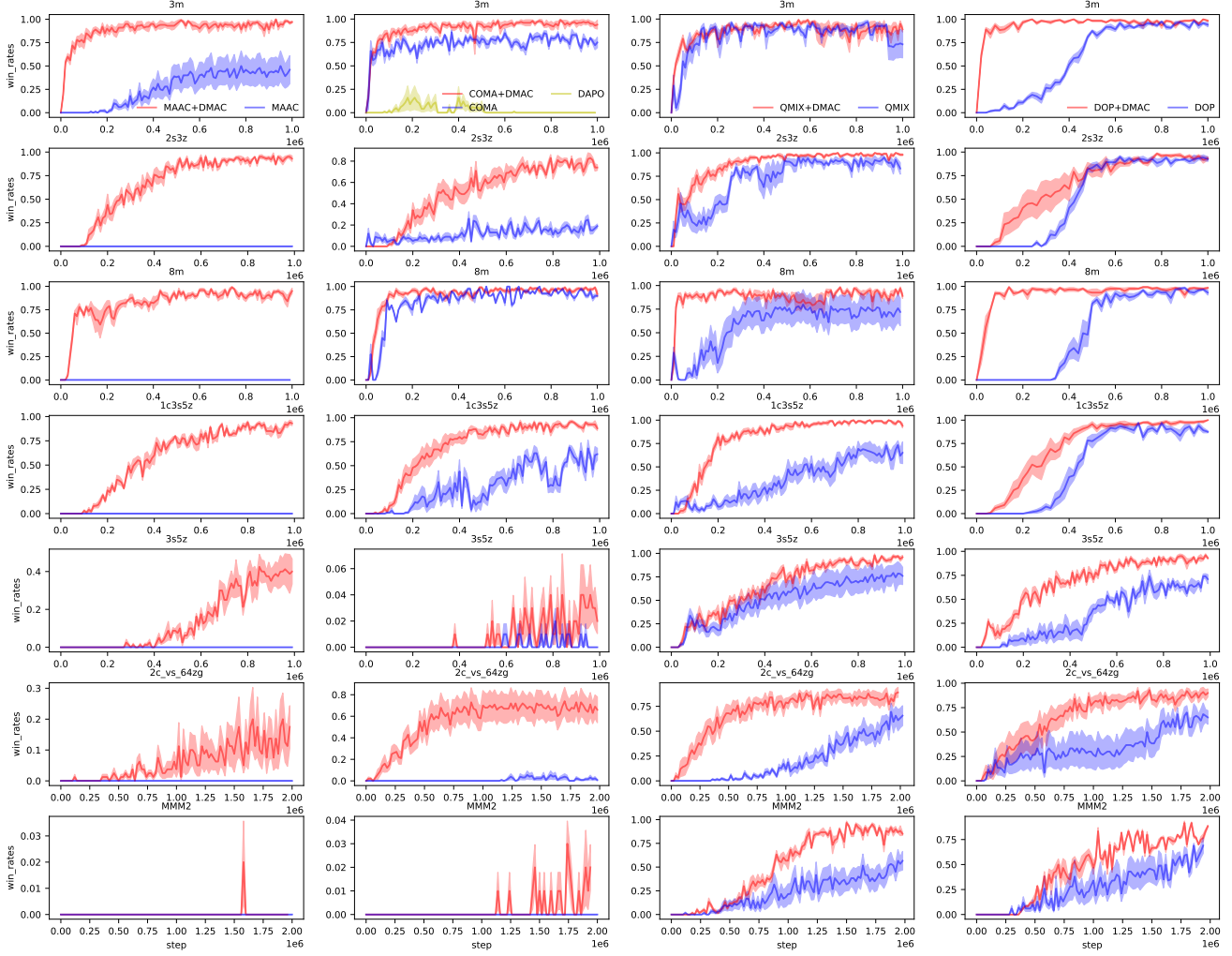
Figure 8. Learning curves in terms of win rate of COMA, MAAC, QMIX, and DOP groups in seven SMAC maps (each row corresponds to a map and each column corresponds to a group).
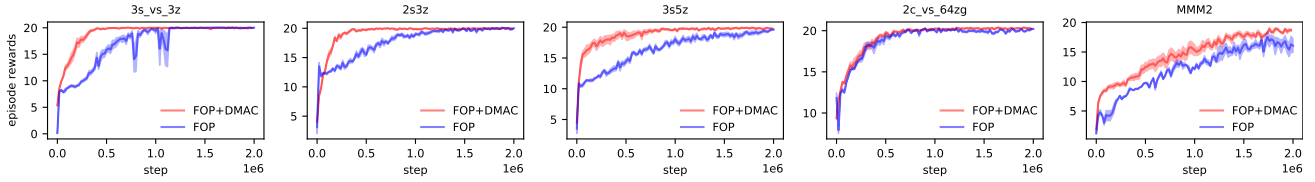

Figure 9. Learning curves in terms of mean episode rewards of FOP+DMAC and FOP in five SMAC maps (each column corresponds to a map).
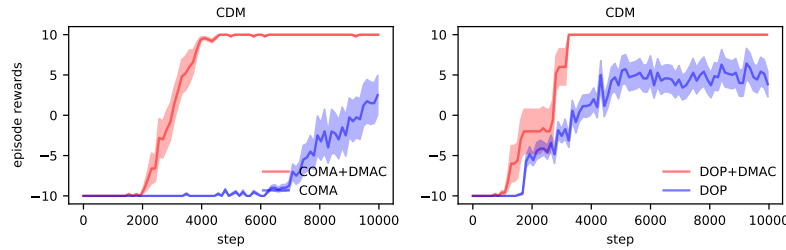

Figure 10. Learning curves in terms of mean episode rewards of COMA and DOP groups in the CDM environment used by the DOP paper.