DEALING WITH NON-STATIONARITY IN MARL VIA TRUST-REGION DECOMPOSITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Non-stationarity is one thorny issue in multi-agent reinforcement learning (MARL). One of the reasons is the policy changes of agents during the learning process. Due to the insufficient theoretical understanding of non-stationarity, the objectives or goals of existing algorithms are inevitably inconsistent and disparate. In this paper, we introduce a novel notion, the δ-$stationarity$ measurement, to explicitly measure the non-stationarity of a policy sequence, which can be further proved to be bounded by the KL-divergence of consecutive joint policies. Since the joint policies' divergence is difficult to accurately estimate, it is highly non-trivial to alleviate the non-stationarity problem by imposing the trust-region constraint on the joint policy. Although it has lower computational complexity to decompose the joint policy and impose trust-region constraints on the factorized policies, simple policy factorization like mean-field approximation will lead to more considerable policy divergence, which can be considered as the trust-region decomposition dilemma. We model the joint policy as a pairwise Markov random field and propose a trust-region decomposition network (TRD-Net) based on message passing to estimate the joint policy divergence more accurately. The **M**ulti-**A**gent **M**irror descent policy algorithm with **T**rust region decomposition, called **MAMT**, is established by adjusting the trust-region of the local policies adaptively in an end-to-end manner. MAMT can approximately constrain the consecutive joint policies' divergence to satisfy δ-stationarity and alleviate the non-stationarity problem. Our method can bring noticeable and stable performance improvement compared with baselines in coordination tasks of different complexity.

1 INTRODUCTION

Multi-agent systems (MAS) can model many real world decision-making problems, such as multi-player games (Berner et al., 2019; Vinyals et al., 2019; Ye et al., 2020), resource allocation (Zimmer et al., 2021; Sheng et al., 2022), and network routing (Mao et al., 2020a;b). In MAS, multiple agents learn and make decisions together in a shared environment, supported by popular MARL algorithms. Although deep RL has achieved great success in single-agent environments, its adaptation to MAS still faces many challenges due to the complicated interactions among agents. This paper focuses on one of these thorny issues, i.e., non-stationarity, caused by changing agents' policies during the learning process. In a partially observable stochastic game (POSG), the state transition function $\mathcal{T}_i$ and the reward function $\mathcal{R}_i$ of agent i depend on the joint action $\boldsymbol{\mathcal{A}}$ of all agents. Policy change of other agents leads to the change of $\mathcal{T}_j$ and $\mathcal{R}_j$ for agent j. Recently, many works have been proposed to deal with this non-stationarity problem. These works can be roughly divided into two categories (Papoudakis et al., 2019): targeted modifications on some standard RL methods' learning schemes (Lowe et al., 2017; Foerster et al., 2018b; Iqbal & Sha, 2019; Baker et al., 2019), and opponent information estimation and sharing (Foerster et al., 2017; Raileanu et al., 2018; Foerster et al., 2018a; Rabinowitz et al., 2018; Al-Shedivat et al., 2018).

Both categories aim to mitigate the negative impact of policy changes to solve the non-stationarity problem. However, due to the insufficient theoretical understanding of non-stationarity issue, the goals and objectives of these algorithms are inevitably inconsistent and disparate, which makes them ineffective in general. Additionally, those methods also require either an unexpansive training scheme or excessive information exchange, which significantly increase the training costs. Recently, some work has shown that naive parameter sharing, a special case of opponent information sharing, can

effectively alleviate the non-stationarity problem (Gupta et al., 2017; Terry et al., 2020). However, we can prove that naive parameter sharing can lead to a suboptimal outcome that is exponentially-worse with the increasing number of agents (more details are described in the appendix).

From the perspective of each agent, solving the non-stationarity problem in MARL can be transformed into solving multiple non-stationary Markov decision processes (MDPs) (Jaksch et al., 2010; Ortner et al., 2020; Cheung et al., 2019; Mao et al., 2021). In these MDPs, *the interactive environment is composed of multiple agents*, and the reason for the non-stationary environment is the changing of agent policies. If the policies of all agents change slowly, the environment can remain stable. In this case, each agent could converge to optimality, i.e., pursue the best response to other agents, which will alleviate the non-stationarity problem. Moreover, the changing opponents will significantly hinder the agent's learning in MAS (Radanovic et al., 2019; Lee et al., 2020; Mao et al., 2021). To address this issue, a popular class of MARL algorithms focus on limiting the divergence between consecutive policies of the learned policy sequence by imposing direct constraints or regularization terms. These methods are referred to as *trust-region-based* or *proximity-based* algorithms (Schulman et al., 2015; 2017; Tomar et al., 2020). However, the precise connection between joint policy divergence and the non-stationarity is still unclear, which motivates us to analyze the relationship theoretically. Furthermore, directly adding the trust-region constraint to the *joint* policy will make the problem intractable. Therefore, effective factorization of the joint policy and trust-region constraint is the key to improve the algorithm's efficiency (Oliehoek et al., 2008).

In this paper, we propose a novel notion called δ-stationarity to measure the stationarity of a given policy sequence. The core of δ-stationarity is the *opponent switching cost*, which is inspired by the *local switching cost* (Bai et al., 2019; Gao et al., 2021) that measure the changing behavior of an single-agent RL, and is used to measures the changing of the agent's joint behavior. Furthermore, the relationship where δ-stationarity is bounded by the KL-divergence of consecutive joint policies is theoretically established. Similar to existing works (Ortner et al., 2020; Cheung et al., 2019; Mao et al., 2021), which use dynamic regret (Jaksch et al., 2010) to measure the optimality of solving non-stationary MDPs algorithms, we can also prove a $\tilde{O}(D_{\max}^{2/3}|\mathcal{O}|^{1/2}\delta_i^{1/4}T)$ dynamic regret bound for each agent. This provides theoretical support to impose the trust-region constraint on the joint policy to alleviate the non-stationarity problem. However, directly dealing with the trust-region constraint on the joint policy is computationally expensive. The natural idea is to decompose the joint policy while imposing trust-region constraints on the factorized policies. However, simple policy factorization like mean-field approximation will lead to sinificant policy divergence, which is considered as the *trust-region decomposition dilemma*.

To this end, we model the joint policy as a Markov random field and propose a trust-region decomposition network (TRD-Net) to adaptively factorize the trust-region of the joint policy and approximately satisfy the δ-stationarity. Local trust-region constraints are imposed on factorized policies, which can be efficiently solved through mirror descent. The TRD-Net constructs the relationship among factorized policies, factorized trust-regions, and the estimated joint policy divergence. To accurately estimate the joint policy divergence, an auxiliary task, which borrowes the idea in offline RL to measure the degree of out-of-distribution, is constructed to train the TRD-Net. The proposed algorithm is denoted as **M**ulti-**A**gent **M**irror descent policy optimization with **T**rust region decomposition, i.e., **MAMT**. MAMT can alleviate the non-stationarity problem and factorize joint trust-region constraint into trust-region constraints on local policies, which could significantly improve the learning effectiveness and robustness. Our contributions mainly consist of the following: 1) We propose a formal non-stationarity definition of MARL, δ-stationarity, which is derived from the local switching cost. δ-stationarity is bounded by the joint policy divergence and bounds the dynamic regret of each agent. 2) The novel trust-region decomposition scheme based on message passing and mirror descent could approximately satisfy the δ-stationarity through a computationally efficient way; 3) Our proposed algorithm could bring noticeable and stable performance improvement in multiple coordination tasks of different complexity than baselines.

2 The Stationarity of the Learning Procedure

To explore and mitigate the non-stationarity in MARL, we first need to model the non-stationarity explicitly. To emphasize, the non-stationarity is not an intrinsic attribute of a POSG but is additionally introduced when using a specific learning algorithm to solve the POSG.

As mentioned above, solving the non-stationarity problem in MARL can be transformed into solving multiple dynamic non-stationary MDPs, where the non-stationary transition probability $\mathcal{T}_i$ and reward function $\mathcal{R}_i$ is caused by the changing of environment (other agents policies). Switching cost is a standard notion in the literature to measure the changing behavior of an single-agent RL algorithm (Cesa-Bianchi et al., 2013; Bai et al., 2019; Gao et al., 2021). We consider the following definition of the (local) switching cost from Bai et al. (2019):

Definition 1 (Local switching cost)**.** *Let H be the horizon of the MDP and K be the number of episodes that the agent can play. The local switching cost (henceforth also "switching cost") between any pair of policies (π, π') is defined as the number of (h, o) pairs on which π and π' are different:*

$$n_{switch}\left(\pi, \pi'\right) := \left|\left\{(h, o) \in [H] \times \mathcal{O} : \pi^h(o) \neq \left[\pi'\right]^h(o)\right\}\right|,$$

where $[H] := 1, \cdots, H$ and $o, \mathcal{O}$ are the local observation and observation space. For an single-agent RL algorithm that employs policies $(\pi^1, \cdots, \pi^K)$, its local switching cost is defined as

$$N_{switch} := \sum_{k=1}^{K-1} n_{switch}\left(\pi_k, \pi_{k+1}\right).$$

In MARL, it needs more attention that the magnitude of the change of other agents' joint policy at each learning step. Thus, we propose the following *opponent switching cost* to measure the changing behavior of opponents by naturally extending the local switching cost:

Definition 2 (Opponent switching cost)**.** *Let H be the horizon of the MDP and K be the number of episodes that the agent can play, so that total number of steps $T := HK$. The opponent switching cost of agent i is defined as the maximum Kullback–Leibler divergence of all $(t, \boldsymbol{o})$ pairs between any pair of opponents' joint policies (π_{-i}, π'_{-i}) on which π_{-i} and π'_{-i} are different:*

$$d^i_{switch}\left(\pi_{-i}, \pi'_{-i}\right) := \max\left\{(t, \boldsymbol{o}) \in [T] \times \mathcal{O} : D_{\mathrm{KL}}\left(\pi^t_{-i}(\boldsymbol{o}) \| \left[\pi'_{-i}\right]^t(\boldsymbol{o})\right)\right\},$$

where $[T] := 1, \cdots, T$ and $\boldsymbol{o}, \mathcal{O}$ are the joint observation and observation space.

The larger the changing magnitude of other agents' joint policy is, the larger the opponent switching cost is, and the more serious non-stationarity problem will be suffered in MARL (Radanovic et al., 2019; Lee et al., 2020; Mao et al., 2021). Therefore, opponent switching cost could play an effective role to measure the non-stationarity. Based on the definition of opponent switching cost, the (non-)stationarity of the learning procedure of each agent and entire MAS can be further defined as follows:

Definition 3 (δ-stationarity of the learning procedure)**.** *For a MAS containing n agents, if we have $d^i_{switch}\left(\pi_{-i}, \pi'_{-i}\right) \leq \delta_i$, then the learning procedure of agent i is δ_i-stationary. Further, if all agents are δ-stationary with corresponding $\{\delta_i\}$, then the learning procedure of entire multi-agent system is δ-stationary with $\delta = \frac{1}{n}\sum_i \delta_i$.*

There is a strong relationship between the Definition 3 and the changing of the $\mathcal{T}_i, \mathcal{R}_i$ for each agent i, which is stated by Lemma 1 and Lemma 2:

Lemma 1. *If any agent i satisfies δ_i-stationarity, then the total variation distance of its two consecutive transition probability distribution $p^t(\cdot|\boldsymbol{o}, a_i)$ and $p^{t+1}(\cdot|\boldsymbol{o}, a_i)$ is bounded by δ_i, i.e., $D_{\mathrm{TV}}\left(p^t(\cdot|\boldsymbol{o}, a_i) \| p^{t+1}(\cdot|\boldsymbol{o}, a_i)\right) \leq 2\ln 2 \cdot \delta_i^{1/2}$.*

Lemma 2. *Assume the absolute value of joint reward is less than 1. If any agent i satisfies δ_i-stationarity, then the total variation distance of its two consecutive reward function $r^t(\boldsymbol{o}, a_i)$ and $r^{t+1}(\boldsymbol{o}, a_i)$ is bounded by δ_i, i.e., $D_{\mathrm{TV}}\left(r^t(\boldsymbol{o}, a_i),\ r^{t+1}(\boldsymbol{o}, a_i)\right) \leq 2\ln 2 \cdot \delta_i^{1/2}$.*

Lemma 1 and Lemma 2 indicate that the δ-stationarity is a reasonable mathematical description of the non-stationarity problem (Hernandez-Leal et al., 2017; Papoudakis et al., 2019; Padakandla, 2021), which can measure the non-stationarity of the learning procedure. Besides, we find that controlling the δ-stationarity of the learning procedure can obtain tighter (dynamic) regret bound for each agent i, which indicates smaller distance to the set of coarse correlated equilibria (CCE) in finite games (Hannan, 2016; Hart & Mas-Colell, 2000; Hsieh et al., 2021). Based on the Lemma 1 and Lemma 2, Theorem 1 states that the (dynamic) regret of each agent i can be bounded by corresponding δ_i (the definitions of $D_{\max}$, B_r and B_p are shown in appendix):

Theorem 1. *Consider the learning procedure of a MAS satisfies the δ-stationarity and each agent satisfies the δ_i-stationarity. Let H be the horizon and K be the number of episodes, so that total number of steps $T := HK$. In addition, suppose that $T \geq B_r + 2D_{max}B_p > 0$, then a $\tilde{O}(D_{\max}^{2/3}|\mathcal{O}|^{1/2}\delta_i^{1/4}T)$ dynamic regret bound is attained for each agent i.*

However, computing the KL-divergence of the consecutive opponents' joint policies $(\pi_{-i}^t, \pi_{-i}^{t+1})$ in Theorem 1 still has high computational complexity. For a MAS with n agents, n constraints on the joint policy divergence should be modeled simultaneously. Before proposing the algorithm, we still need to relax the constraints on the joint policy divergence to increase efficiency. Based on the following theorem, the constraints on the joint policy divergence of the other agents can be limited by the joint policy divergence of all agents, independent of the agent number. Both detailed proofs of Lemma 1, Theorem 1 and the following Theorem 2 can be found in the appendix.

Theorem 2. *For a multi-agent system, the maximum KL-divergence of all agents' consecutive joint policies (π, π') is the upper bound of the average opponent switching cost*

$$(1/n) \cdot \sum_{i=1}^{n} d_{switch}^{i}\left(\pi_{-i}, \pi'_{-i}\right) \leq \max_{\boldsymbol{o}} D_{\mathrm{KL}}\left(\pi(\cdot|\boldsymbol{o}) \| \pi'(\cdot|\boldsymbol{o})\right).$$

With the above Theorem 2, it only needs to impose only one trust-region constraint on all agents' joint policy to control the non-stationarity of the entire learning procedure. This allows us to design a MARL algorithm with better effectiveness to approximate the defined δ-stationarity.

3 The Proposed MAMT Method

According to Theorem 2, we need to properly constrain the maximum divergence of consecutive joint policies to make the learning procedure more stable and efficient. In other words, the algorithm needs to balance between eliminating the non-stationarity (i.e., $\delta \to 0$) and fast learning (i.e., $\delta \to \infty$). To emphasize, based on Definition 3, Theorem 1 and Theorem 2, we can modify the original MARL problem model by adding stationarity constraint. This problem imposes a constraint that the KL divergence is bounded at every point in the state space. While it is motivated by the theory, this problem is impractical to solve due to the large number of constraints. Instead, we can use a heuristic approximation, similar as Schulman et al. (2015), which considers the average KL divergence:

$$\begin{aligned} \boldsymbol{\pi}^{k+1} \in \arg\max_{\boldsymbol{\pi}} & \sum_{i=1}^{n} V_i^{\boldsymbol{\pi}}(\boldsymbol{o}), \quad \forall \boldsymbol{o}, \\ \text{s.t.} \quad & \mathbb{E}_{\boldsymbol{o}\sim\mathcal{D}}\left[\mathbb{E}_{\boldsymbol{a}\sim\boldsymbol{\pi}}\left[D_{\mathrm{KL}}\left(\pi(\boldsymbol{a} \mid \boldsymbol{o}), \pi^k(\boldsymbol{a} \mid \boldsymbol{o})\right)\right]\right] \leq \delta, \end{aligned} \tag{1}$$

where $\boldsymbol{\pi}$ represents the joint policy of all agents; $\boldsymbol{o} := (o_1, \cdots, o_i, \cdots, o_n)$ represents the joint observation with n be the agent number. $\mathcal{D}$ is the replay buffer. Considering the low sample efficiency of MARL, we did not adapt the fully on-policy learning, but used a smaller replay buffer. In this way, it can be ensured that samples are less different from the current policy. Directly imposing a constraint on the joint policy as in (1) will make the problem significantly more difficult and computationally intractable. A straightforward way is to evenly distribute the trust-region to all agents based on mean-field policy approximation (as shown in Figure 6 in the appendix). However, this will bring up the problem which we call *trust-region decomposition dilemma*.

3.1 Trust-region Decomposition Dilemma

Formally, inspired by Lowe et al. (2017), Foerster et al. (2018b) and Iqbal & Sha (2019), we first introduce the *mean-field approximation* assumption.

Assumption 1 (Mean-Field Approximation)**.** *We use the mean-field variation family $\boldsymbol{\pi}$ to estimate the joint policy of all agents, i.e., $\boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{o}) = \prod_{i=1}^{n} \pi_i(a_i|o_i)$.*

Based on mean-field approximation assumption, the joint policy trust-region constraint can be factorized into trust-region constraints of local policies with the following theorem.

Theorem 3. *For any consecutive policies which are belong to mean-field variational family $\boldsymbol{\pi}$ and $\boldsymbol{\pi}^k$ in the policy sequence obtained by any learning algorithm, the KL divergence constraint on joint policy in (1) is equivalent with the following summation constraints of local policies, i.e.,*

$$\sum_{i=1}^{n} \mathbb{E}_{o_i\sim\mathcal{D}}\left[\mathbb{E}_{a_i\sim\pi_i}\left[D_{\mathrm{KL}}(\pi_i(a_i|o_i), \pi_i^k(a_i|o_i))\right]\right] \leq \delta. \tag{2}$$

Further the joint summation trust-region constraint (2) can be equivalently decomposed into the following local trust-region constraints, i.e.,

$$\mathbb{E}_{o_i\sim\mathcal{D}}\left[\mathbb{E}_{a_i\sim\pi_i}\left[D_{\mathrm{KL}}(\pi_i(a_i|o_i),\pi_i^k(a_i|o_i))\right]\right]\leq\delta_i,\quad\forall i, \tag{3}$$

with $\sum_{i=1}^n\delta_i=\delta$, $0\leq\delta_i\leq\delta$; π_i *and* π_i^k *represent the consecutive local policies of agent* i; o_i *and* u_i *represent the local observation and initial local observation distribution of agent* i *respectively.*

Due to the summation constraint term (2), it is still need to jointly train the policies of all agents with limited effectiveness and solvability. Therefore, we further equivalently transform (2) into (3), which is proposed to constraint each agent's local policy separately*. Based on Theorem 3, problem (1) can be reformulated as

$$\boldsymbol{\pi}^{k+1}\in\arg\max\textstyle\sum_{i=1}^n V_i^{\boldsymbol{\pi}}(\boldsymbol{o}),\forall\boldsymbol{o},\text{s.t. }\mathbb{E}_{o_i\sim\mathcal{D}}\left[\mathbb{E}_{a_i\sim\pi_i}\left[\mathrm{KL}\left(\pi_i\left(a_i\mid o_i\right),\pi_i^k\left(a_i\mid o_i\right)\right)\right]\right]\leq\delta/n,\forall i. \tag{4}$$

The trust-region decomposition scheme adopted in the above, we called MAMD, is based on the mean-field approximation assumption (i.e., formulation (4)). The similar optimization problem is also obtained by Li & He (2020) that try to implement TRPO for MARL through distributed consensus optimization. However, decomposing the trust-region inappropriately will make the algorithm converge to sub-optimal solutions, which can be numerically proved through the following example. By considering a simple MAS with three agents i,j,k, we assume that the agent k is independent with two agents i and j. Then for the multi-agent system, we have $p_i(\boldsymbol{o}'|\boldsymbol{o},a_i,a_j,a_k)=p_i(\boldsymbol{o}'|\boldsymbol{o},a_i,a_j),p_j(\boldsymbol{o}'|\boldsymbol{o},a_j,a_i,a_k)=p_j(\boldsymbol{o}'|\boldsymbol{o},a_j,a_i),p_k(\boldsymbol{o}'|\boldsymbol{o},a_k,a_i,a_j)=p(\boldsymbol{o}'|\boldsymbol{o},a_k)$.

The change of agent k's policy cannot affect the state transition probability of i and j, and vice versa. As a result, the constraints on the agent i and agent j are "insufficient" but the constraint for agent k is "excessive", when we decompose the trust-region δ in (2) into three parts equally, i.e., $\delta/3$. This phenomenon is called *trust-region decomposition dilemma*† in this paper. Theoretically, the reason for the trust-region decomposition dilemma might be the inaccurate estimation of the joint policy divergence based on the mean-field approximation assumption. The simple summation of local policies' divergence is not equal to the divergence of the joint policy, while the gap might be enormous. Once the joint policy divergence cannot be accurately estimated, the non-stationarity cannot be judged accordingly, and imposing constraints on it may run counter to our goals.

3.2 The MAMT Algorithm Framework

To achieve more reasonable decomposition, recent works (Böhmer et al., 2020; Qu et al., 2020; Li et al., 2021) modeled the joint policy as a Markov random field with pairwise interactions (shown in the Figure 6 in the appendix) based on graph neural networks (GNN). Inspired by these methods, a similar mechanism is utilized in this paper to decompose the joint policy and the trust-region. The joint policy divergence is related to the current factorized policies and the factorized trust-regions, which can be input to the GNN. Therefore, we can appropriately decompose the trust-region in an end-to-end procedure by minimizing the modeled joint policy's divergence. The employed GNN is denoted as *trust-region decomposition network* and the network structure is shown in Figure 1.

However, two critical issues need to be adequately solved to guarantee an implementable and tractable algorithm. For the reason that the joint policy's divergence cannot be directly calculated, a surrogate is needed as the supervision signal to train the TRD-Net; Furthermore, the dependence between agents is an essential factor affecting the joint policy divergence, and we need to model the dependencies between agents reasonably. In the following, we will solve these problems in a targeted way and propose our MAMT algorithm framework. MAMT uses the TRD-Net to estimate the joint policy divergence more accurately. At the same time, the trust-region sizes of the local policies are adaptively adjusted by the trade-off between minimizing joint policy divergence and maximizing accumulated reward, thereby solving the trust-region decomposition dilemma.

Improved Joint Policy Divergence Approximation. Instead of directly calculating the joint policy divergence with the summation of local policy divergences, we estimate it through the TRD-Net. TRD-Net needs to be learned based on some supervised information, which can be obtained by

*In this case, the number of constraints will increase to n, however the number of constraints will become n^2 if the decomposition technique is imposed on the n joint policy of the other agents.

†We verify the existence of trust-region decomposition dilemma through a typical example in the appendix.

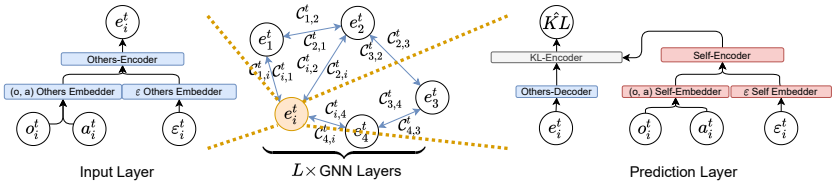


Figure 1: The architecture of trust-region decomposition network. "MP" denotes message passing. Each agent i first encodes its local observation o_i^t and action a_i^t, and concatenates them with the embedding of the local trust-region coefficient, and finally gets the agent's embedding from the MP-Encoder output (see the left Input Layer). Next, we construct all agents as a weighted undirected graph, the weights $C_{*,*}^t$ are calculated in advance, and the updated agent's embedding is obtained through graph convolution (see the middle the GNN Layers). At the same time, similar to the input layer, each agent obtains another agent's embedding through another neural network with the same structure but different parameters. Finally, we concatenate the two embeddings together and input them into KL-Encoder to estimate the current joint policy divergence (see the right Prediction Layer).

a well-defined indicator of non-stationarity. As discussed before, the non-stationarity problem is caused by the dynamic changing of the other agents' policies, which leads to the out-of-distribution of sampled data from the environment. As a result, we borrowed the idea in offline RL to measure the degree of out-of-distribution through the model-based RL (Kidambi et al., 2020; Yu et al., 2020), with the purpose to reasonably characterize the non-stationarity. Specifically, each agent i has a prediction model $h_{\phi_i^j}$ for each other agent j, predicting the other agent's actions based on its local observation. The summation of the divergence between the predicted action distribution and the actual action distribution of all other agents could represent the non-stationarity of the agent i, i.e.,

$$\mathcal{D}_{i,\mathrm{ns}}^t = \Pi_{\mathrm{ns}}\left(\sum_{j\neq i}\mathcal{C}_{i,j}^t\left(\mathrm{KL}\left[h_{\phi_i^j}(o_i^t)\|\pi_{\psi_j}(o_j^t)\right]\right)\right), \tag{5}$$

where "ns" denotes the "non-stationarity"; $\mathcal{C}_{i,j}^t$ is the coordination coefficient (we will explain soon) between agent i and agent j; π_{ψ_j} is the policy of agent j which is parameterized by ψ_j; Π_{ns} represents the projection function, which constrains $\mathcal{D}_{i,\mathrm{ns}}^t$ to a specific range. Kim et al. (2020) also introduces inter-agent action prediction, but Kim et al. (2020) is to promote collaboration, and this paper is to estimate the non-stationarity of the joint policy better. Furthermore, the non-stationarity of MAS can be denoted as the summation of all local non-stationarities, i.e.,$\mathcal{D}_{\mathrm{ns}}^t = \sum_i \mathcal{D}_{i,\mathrm{ns}}^t$, which can be considered as the supervised signal for trust-region decomposition network training. The loss function to learn the trust-region decomposition network can be formulated as

$$\min_\theta \mathcal{L}_{\mathrm{ns}} = \left(\sum_i \hat{\mathrm{KL}}_i^t - \sum_i \mathcal{D}_{i,\mathrm{ns}}^t\right)^2, \tag{6}$$

where θ and $\hat{\mathrm{KL}}_i^t$ are the parameters and the output of the trust-region decomposition network.

Calculation of Coordination Coefficient. We use a non-negative real number, coordination coefficient, to represent the dependency between two agents[‡]. In the learning procedure, the coordination relationship between agents is changing with the learning of policies and completing tasks. Therefore, we model the cooperative relationship between two agents based on counterfactual (Foerster et al., 2018b; Jaques et al., 2019). In Jaques et al. (2019), the causal influence reward of agent i w.r.t. opponent j is calculated by $c_{i,j}^t = D_{KL}\left[p\left(a_j^t \mid a_i^t, o_j^t\right)\|p\left(a_j^t \mid o_j^t\right)\right]$, where $p\left(a_j^t \mid s_j^t\right) = \sum_{\tilde{a}_i^t} p\left(a_j^t \mid \tilde{a}_i^t, o_j^t\right) p\left(\tilde{a}_i^t \mid o_j^t\right)$. It can be seen that the greater the causal influence reward, the tighter the coordination between the two agents. However, when calculating the causal influence reward, the agent's policy needs to be modified. That is, the agent needs to explicitly rely on the actions of other agents when making decisions. To solve this problem, we modified the process of calculating the counterfactual baseline in Foerster et al. (2018b) based on the idea of Jaques et al. (2019), and obtained a new way of calculating the coordination coefficient

$$\hat{\mathcal{C}}_{i,j} = \mathrm{softmax}\left(\left|Q_i(\mathbf{a}_{\setminus j}) - Q_i(\mathbf{a})\right|\right), \ \mathcal{C}_{i,j} = \begin{cases} \hat{\mathcal{C}}_{i,j}, & \hat{\mathcal{C}}_{i,j} \geq \delta, \\ 0, & \hat{\mathcal{C}}_{i,j} \leq \delta, \end{cases} \tag{7}$$

where $Q_i(\cdot)$ is the centralized critic of agent i. $\mathbf{a}$ is the joint action of all agents, and $\mathbf{a}_{\setminus j}$ is the joint action of all agents expect for j; δ is the threshold to keep sparsity.

[‡]In this paper, we only model the pairwise relationship.

Next we formally propose our **M**ulti-**A**gent **M**irror descent policy algorithm with **T**rust region decomposition, i.e., MAMT. The learning of local trust-region parameter δ_i involves the mutual balance of two parts, i.e., the non-stationarity of the learning procedure and the performance of all agents. The objective of δ_i learning is to maximize

$$\mathcal{F}(\boldsymbol{\delta}) := \mathbb{E}_{\boldsymbol{o}\sim\mu}\left[\sum_{i=1}^{n} V_i^{\boldsymbol{\pi}(\boldsymbol{\delta})}(\boldsymbol{o}) - \hat{\mathrm{KL}}_i^{\boldsymbol{\pi}(\boldsymbol{\delta})}(\boldsymbol{o}, \boldsymbol{\delta}; \theta)\right],$$

where $\boldsymbol{\delta} = \{\delta_i\}_{i=1}^n$ and $\boldsymbol{\pi}(\boldsymbol{\delta})$ represent the joint policy is related to the δ_i of all agents; $\hat{\mathrm{KL}}_i^{\boldsymbol{\pi}(\boldsymbol{\delta})}$ denotes the output of the TRD-Net which is parameterized by θ. Finally, the learning of $\{\delta_i\}$ and θ can be modeled as a bilevel optimization problem (Dempe & Zemkoho, 2020)

$$\boldsymbol{\delta}^\star = \arg\max \;\; \mathcal{F}(\boldsymbol{\delta}, \theta^\star(\boldsymbol{\delta})), \quad \text{s.t. } \theta^\star(\boldsymbol{\delta}) = \arg\min \;\; \mathcal{L}_{\mathrm{ns}}(\boldsymbol{\delta}, \theta).$$

We can employ the efficient two-timescale gradient descent method to simultaneously perform gradient update for both $\boldsymbol{\delta}$ and θ. Specifically, we have

$$\boldsymbol{\delta}_{k+1} \leftarrow \boldsymbol{\delta}_k - \alpha_k \cdot \nabla_{\boldsymbol{\delta}} \mathcal{F}(\boldsymbol{\delta}_k, \theta_k), \; \theta_{k+1} \quad \leftarrow \theta_k - \beta_k \cdot \nabla_\theta \mathcal{L}_{\mathrm{ns}}(\boldsymbol{\delta}_k, \theta_k), \; s.t. \; \alpha_k/\beta_k \to 0, \tag{8}$$

where $\frac{\alpha_k}{\beta_k} \to 0$ indicates that $\{\theta_k\}_{k\geq 0}$ updates faster than $\{\boldsymbol{\delta}_k\}_{k\geq 0}$. In practical, we make α_k and β_k equal but perform more gradient descent steps on θ, similar as Fujimoto et al. (2018).

By directly combing the MAAC (Iqbal & Sha, 2019) and the mirror descent technique in MDPO (Tomar et al., 2020), we can get the algorithm framework for solving problem formulated in Equation (1) after decomposing the trust-region constraint. Each agent i has its local policy network π_{ψ_i} and a local critic network Q_{ζ_i}, which is similar to MADDPG[§] (Lowe et al., 2017). All critics are centralized updated iteratively by minimizing a joint regression loss function, i.e.,

$$\mathcal{L}_{critic}(\zeta_1, \cdots, \zeta_n) = \sum_{i=1}^{n} \mathbb{E}_{(o,a,r,o')\sim\mathcal{D}}\left[\left(Q_{\zeta_i}(o,a) - y_i\right)^2\right], \tag{9}$$

with $y_i = r_i + \gamma \mathbb{E}_{a'\sim\pi_{\bar{\psi}}(o')}[Q_{\bar{\zeta}_i}(o', a') - \alpha \log(\pi_{\psi_i}(a_i' \mid o_i'))]$, and the parameters of the attention module are shared among agents; $\pi_{\bar{\psi}}$ represents the joint target policy (similar as target Q-network in DQN) of all agents; $Q_{\bar{\zeta}_i}$ represents the target local critic of agent i. α denotes the temperature balancing parameter between maximal entropy and rewards.

As for the individual policy updating step, the mirror descent technique (Tomar et al., 2020) is introduced to deal with the decomposed local trust-region constraints. The policy gradient is calculated with the following form:

$$\nabla_{\psi_i} J(\pi_\psi) = \mathbb{E}_{o\sim D, a\sim\pi}\left[\nabla_{\psi_i} \log(\pi_{\psi_i}(a_i \mid o_i)) \left(\mathrm{KL}[\theta_i \| \theta_i^{\mathrm{old}}]/\delta_i + Q_{\zeta_i}(o,a) - b(o, a_{\setminus i})\right)\right],$$

where $b(o, a_{\setminus i})$ denotes the counterfactual baseline (Foerster et al., 2018b); $\pi_{\theta_i^{\mathrm{old}}}$ represents the policy of agent i at last step; δ_i is the assigned trust-region range; $\mathrm{KL}[\theta_i \| \theta_i^{\mathrm{old}}] = \log(\pi_{\theta_i}(a_i \mid o_i)) - \log(\pi_{\theta_i^{\mathrm{old}}}(a_i \mid o_i))$. However, if we only perform a single-step SGD, the resulting gradient would be equivalent to vanilla policy gradient and misses the purpose of enforcing the trust-region constraint. As a result, the policy update at each iteration k involves m SGD steps as

$$\psi_{i,k}^{(0)} = \psi_{i,k}, \quad j = 0, \ldots, m-1; \quad \psi_{i,k}^{(j+1)} \leftarrow \psi_{i,k}^{(j)} + \eta \nabla_{\psi_i} J\left(\pi_\psi\right)\big|_{\psi_i = \psi_{i,k}^{(j)}}, \quad \psi_{i,k+1} = \psi_{i,k}^{(m)}.$$

For off-policy mirror descent policy optimization, performing multiple SGD steps at each iteration becomes increasingly time-consuming as the value of m grows, Therefore, similar with Tomar et al. (2020), we resort to staying close to an m step old copy of the current policy while performing a single gradient update at each iteration of the algorithm. This copy is updated every m iterations with the parameters of the current policy. The pseudo-code of MAMT is shown in the appendix.

4 Experiments

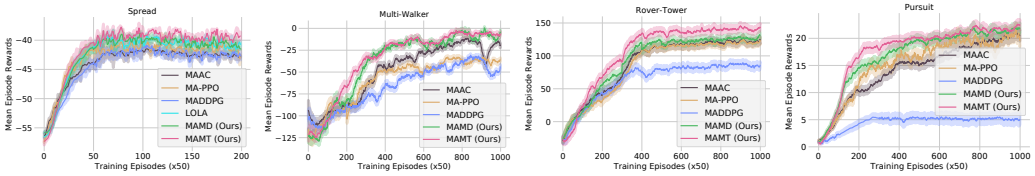
Our experiments aim to verify the effectiveness and robustness of the MAMT algorithm. The effectiveness of the trust-region constraints on joint policy, the existence of the trust-region decomposition dilemma, and the capacity of the trust-region decomposition will be discussed and proved. Four

[§]Our algorithm also follows the centralized critics and decentralized actors framework.

coordination environments with increasing complexity, i.e. *Spread*, *Multi-Walker*, *Rover-Tower*, *Pursuit* (more details can be found in appendix) are employed.

Baseline Algorithms. **MADDPG** (Lowe et al., 2017), which is one of the typical CCDA algorithms, is chosen as one baseline. The core of our proposed methods is the proposed trust-region techniques, which motivates us to set the combination of PPO and MADDPG as another baseline and be labeled as **MA-PPO** (Yu et al., 2021). The **MAAC** (Iqbal & Sha, 2019) is considered the baseline, which is based on the attention mechanism to modeling other agents. MA-PPO's policy network also uses an attention mechanism similar to MAAC for fairness. In addition, the **LOLA**[¶] (Foerster et al., 2018a) algorithm that models opponent agents to solve the non-stationarity problem is compared as the baseline. Since LOLA employed the Taylor expansion to explicitly add the information of the opponents to the gradient, while the extension to more agents seems to be difficult and it is only compared in the Spread environment. To emphasize, MAMT without the TRD-Net, which we called **MAMD** algorithm, is also set to be the baseline. In the MAMD algorithm, the trust-region constraint is evenly distributed to the local policies of all agents.

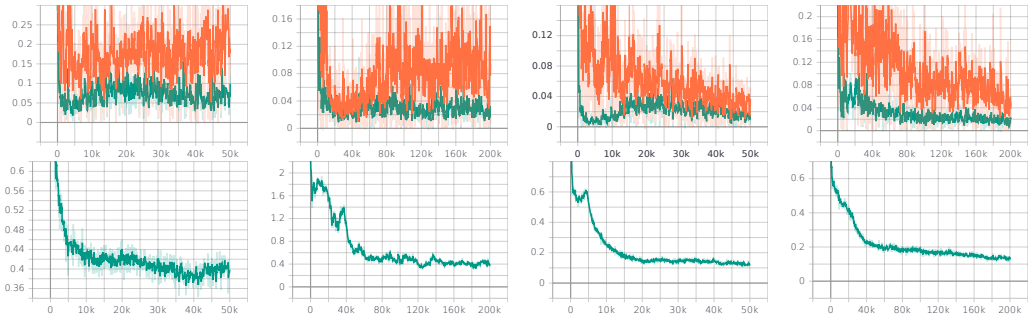
Comparisons. we compare MAMT with all baselines and get the following three conclusions:



(a) Spread. (b) Multi-Walker. (c) Rover-Tower. (d) Pursuit.

Figure 2: The performance comparisons under coordination tasks with different complexity.

1). Combine the single-agent trust-region technique with MARL directly cannot bring stable performance improvement. The averaged episode rewards of all methods in four environments are shown in Figure 2. From the comparison of MAAC and MA-PPO, it can be seen that directly combining the single-agent trust-region technique with MARL cannot bring stable performance improvement. In the *Rover-Tower* and *Pursuit*, MA-PPO did not bring noticeable improvement in final performance; MA-PPO only has a subtle improvement in the simple *Spread* environment and even causes performance degradation in *Multi-Walker*. In contrast, MAMT can bring noticeable and stable performance improvements in all environments. It is worth noting that MAMD also shows promising results. This shows that even a simple trust-region decomposition can bring noticeable improvement in simple scenarios. This further verifies the rationality and effectiveness of non-stationarity modeling.



(a) Spread. (b) Multi-Walker. (c) Rover-Tower. (d) Pursuit.

Figure 3: **Upper:** The averaged KL divergence of each agent with the orange and green line representing MAAC and MAMT respectively; **Bottom:** The $\mathcal{D}_{\text{ns}}$ of all agents.

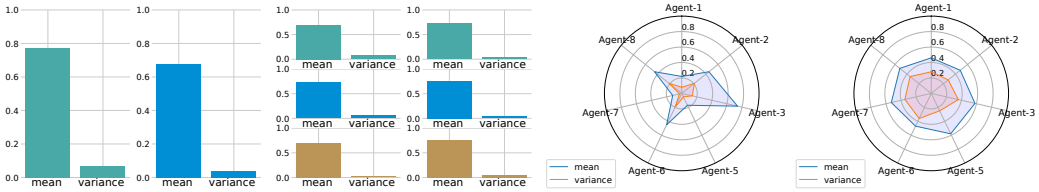
2). MAMT achieves the trust-region constraint on the local policies and minimizes the non-stationarity of the learning procedure. Below we conduct an in-depth analysis of the reasons why MAMT can noticeably and stably improve the final performance. MAMT uses an end-to-end approach to adaptively adjust the sizes of the trust-regions of the local policies according to the current

[¶]We combine LOLA with DQN to solve more complex stochastic game.

dependencies between agents to constrain the KL divergence of the joint policy more accurately. Therefore, we analyze MAMT from the two perspectives: the trust-region constraint satisfaction of the local policies and the non-stationarity satisfaction via surrogate measure. Figure 3 shows the average value of the KL divergence of all agents' local policies during learning procedures in different environments. It can be seen from the figure that MAMT controls the policy divergence. The non-stationarity of the learning procedure is challenging to calculate according to the definition, so we use the surrogate measure $\mathcal{D}_{\text{ns}}$ to represents the non-stationarity. It can be seen from Figure 3 that as the learning progresses, the non-stationarity is gradually decreasing.

(a) Spread. (b) Multi-Walker. (c) Rover-Tower. (d) Pursuit.

Figure 4: Mean and variance of local trust-region of each agent. The y-axis range from 0.01 to 100.0.



(a) Spread. (b) Multi-Walker. (c) Rover-Tower. (d) Pursuit.

Figure 5: Mean and variance of pairwise coordination coefficients. Different colors represent different agents in (a) and (b). In addition, since there are fewer agents in *Spread* and *Multi-Walker*, the display effect of the radar chart is poor, so we use the histogram for a more precise display in (a) and (b).

3). Trust-region decomposition works better when there are more agents and more complex coordination relationships. Figure 2 shows that MAMT and MAMD have similar performance in the *Spread* and *Multi-Walker*, but MAMT is noticeably better than MAMD in the *Rover-Tower* and *Pursuit*. To analyze the reasons for this phenomenon, we separately record the coordination coefficients between agents in the learning procedure and the sizes of the trust-regions of the local policies, see Figure 5 and Figure 4. It can be seen from Figure 5 that the *Spread* and *Multi-Walker* are relatively simple, and the number of agents is small. This makes the interdependence between agents is very tight (larger mean) and will not change over time (variance is microscopic). Moreover, the corresponding trust-region sizes are similar to coordination coefficients. These figures show that in simple coordination tasks with a small number of agents, trust-region decomposition is not a severe problem, which leads to the fact that the MAMT cannot bring a noticeable performance improvement. In the other two environments, the dependence between agents is not static (the variance in Figure 5 is more considerable) due to the more significant number of agents and more complex tasks. This also makes the sizes of the trust-regions of the agent's local policies fluctuate significantly in different stages (the variance in Figure 4 is more considerable). MAMT alleviates the trust-region decomposition dilemma through the TRD-Net, which brings a noticeable and stable performance improvement.

5 Conclusion

In this paper, we define the δ-stationarity to explicitly model the stationarity in the learning procedure of the multi-agent system and provide a theoretical basis for proposing an achievable learning algorithm based on joint policy trust-region constraints. To solve the trust-region decomposition dilemma caused by the mean-field approximation and to estimate the joint policy divergence more accurately, we propose an efficient and robust algorithm MAMT combining message passing and mirror descent with the purpose to satisfy δ-stationarity. Experiments show that MAMT can bring noticeable and stable performance improvement and more suitable for large-scale scenarios with complex coordination relationships between agents.

Ethics Statement. Our method is not a generative model, nor does it involve the training of super-large-scale models. The training data is sampled in the simulated environments, so it does not involve fairness issues. Our method also does not involve model or data stealing and adversarial attacks.

Reproducibility Statement. The source code of this paper is available at https://anonymous.4open.science/r/MAMT. We specify all the training details (e.g., hyperparameters, how they were chosen), the error bars (e.g., with respect to the random seed after running experiments multiple times), and the total amount of compute and the type of resource used (e.g., type of GPUs and CPUs) in the Section 4 or the Appendix F. We also cite the creators of partial code of our method and mention the license of the assets in the Appendix F.

REFERENCES

Maruan Al-Shedivat, Trapit Bansal, Yura Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. Continuous adaptation via meta-learning in nonstationary and competitive environments. In *ICLR*, 2018.

Yu Bai, Tengyang Xie, Nan Jiang, and Yuxiang Wang. Provably efficient q-learning with low switching cost. In *NeurIPS*, 2019.

Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. In *ICLR*, 2019.

Christopher Berner, G. Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, D. Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, R. Józefowicz, Scott Gray, C. Olsson, Jakub W. Pachocki, M. Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, J. Schneider, S. Sidor, Ilya Sutskever, Jie Tang, F. Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning. *ArXiv*, abs/1912.06680, 2019.

Wendelin Böhmer, Vitaly Kurin, and Shimon Whiteson. Deep coordination graphs. In *ICML*, 2020.

Nicolo Cesa-Bianchi, Ofer Dekel, and Ohad Shamir. Online learning with switching costs and other adaptive adversaries. In *NeurIPS*, 2013.

Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. Non-stationary reinforcement learning: The blessing of (more) optimism. *Machine Learning eJournal*, 2019.

Stephan Dempe and Alain Zemkoho. *Bilevel optimization*. Springer, 2020.

Jakob Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. In *AAMAS*, 2018a.

Jakob N. Foerster, Nantas Nardelli, Gregory Farquhar, Triantafyllos Afouras, Philip H. S. Torr, Pushmeet Kohli, and Shimon Whiteson. Stabilising experience replay for deep multi-agent reinforcement learning. In *ICML*, 2017.

Jakob N Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *AAAI*, 2018b.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *ICML*, 2018.

Minbo Gao, Tianle Xie, Simon S Du, and Lin F Yang. A provably efficient algorithm for linear markov decision process with low switching cost. *ArXiv preprint ArXiv:2101.00494*, 2021.

Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *AAMAS*, 2017.

James Hannan. Chapter 4. Approximation to rayes risk in repeated play. In *Contributions to the Theory of Games (AM-39), Volume III*, pp. 97–140. Princeton University Press, 2016.

Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

Pablo Hernandez-Leal, Michael Kaisers, Tim Baarslag, and Enrique Munoz de Cote. A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv preprint arXiv:1707.09183*, 2017.

Yu-Guan Hsieh, Kimon Antonakopoulos, and Panayotis Mertikopoulos. Adaptive learning in continuous games: Optimal regret bounds and convergence to nash equilibrium. *arXiv preprint arXiv:2104.12761*, 2021.

Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. In *ICML*, 2019.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(51):1563–1600, 2010.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *ICML*, 2019.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. MOReL: Model-based offline reinforcement learning. In *NeurIPS*, 2020.

Woojun Kim, Whiyoung Jung, Myungsik Cho, and Youngchul Sung. A maximum mutual information framework for multi-agent reinforcement learning. *arXiv preprint arXiv:2006.02732*, 2020.

Chung-wei Lee, Haipeng Luo, Chen-Yu Wei, and Mengxiao Zhang. Linear last-iterate convergence for matrix games and stochastic games. *ArXiv*, abs/2006.09517, 2020.

Hepeng Li and Haibo He. Multi-agent trust region policy optimization. *arXiv preprint arXiv:2010.07916*, 2020.

Sheng Li, Jayesh K Gupta, Peter Morales, Ross Allen, and Mykel J Kochenderfer. Deep implicit coordination graphs for multi-agent reinforcement learning. In *AAMAS*, 2021.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *NeurIPS*, 2017.

Hangyu Mao, Wulong Liu, Jianye Hao, Jun Luo, Dong Li, Zhengchao Zhang, Jun Wang, and Zhen Xiao. Neighborhood cognition consistent multi-agent reinforcement learning. In *AAAI*, 2020a.

Hangyu Mao, Zhengchao Zhang, Zhen Xiao, Zhibo Gong, and Yan Ni. Learning multi-agent communication with double attentional deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 34(1):1–34, 2020b.

Weichao Mao, Kaiqing Zhang, Ruihao Zhu, David Simchi-Levi, and Tamer Başar. Near-optimal model-free reinforcement learning in non-stationary episodic mdps. In *ICML*, 2021.

Frans A Oliehoek, Matthijs TJ Spaan, and Nikos Vlassis. Optimal and approximate Q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

Ronald Ortner, Pratik Gajane, and Peter Auer. Variational regret bounds for reinforcement learning. In *UAI*, 2020.

Sindhu Padakandla. A survey of reinforcement learning algorithms for dynamically varying environments. *ACM Computing Surveys (CSUR)*, 54(6):1–25, 2021.

Georgios Papoudakis, Filippos Christianos, A. Rahman, and Stefano V. Albrecht. Dealing with non-stationarity in multi-agent deep reinforcement learning. *ArXiv*, abs/1906.04737, 2019.

Chao Qu, Hui Li, Chang Liu, Junwu Xiong, James Zhang, Wei Chu, Yuan Qi, and Le Song. Intention propagation for multi-agent reinforcement learning. *arXiv preprint arXiv:2004.08883*, 2020.

Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew Botvinick. Machine theory of mind. In *ICML*, 2018.

Goran Radanovic, Rati Devidze, David Parkes, and Adish Singla. Learning to collaborate in markov decision processes. In *ICML*, 2019.

Roberta Raileanu, Emily Denton, Arthur Szlam, and Rob Fergus. Modeling others using oneself in multi-agent reinforcement learning. In *ICML*, 2018.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *ICML*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Junjie Sheng, Yiqiu Hu, Wenli Zhou, Lei Zhu, Bo Jin, Jun Wang, and Xiangfeng Wang. Learning to schedule multi-numa virtual machines via reinforcement learning. *Pattern Recognition*, 121: 108254, 2022.

Justin K Terry, Nathaniel Grammel, Ananth Hari, Luis Santos, and Benjamin Black. Revisiting parameter sharing in multi-agent deep reinforcement learning. *arXiv preprint arXiv:2005.13625*, 2020.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. *arXiv preprint arXiv:2005.09814*, 2020.

Oriol Vinyals, I. Babuschkin, W. Czarnecki, Michaël Mathieu, Andrew Dudzik, J. Chung, D. Choi, R. Powell, Timo Ewalds, P. Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, L. Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, A. S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, D. Budden, Yury Sulsky, James Molloy, T. L. Paine, Caglar Gulcehre, Ziyu Wang, T. Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, O. Smith, T. Schaul, T. Lillicrap, K. Kavukcuoglu, Demis Hassabis, Chris Apps, and D. Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, 2019.

Deheng Ye, Guibin Chen, W. Zhang, Sheng Chen, Bo Yuan, B. Liu, J. Chen, Z. Liu, Fuhao Qiu, Hongsheng Yu, Yinyuting Yin, Bei Shi, L. Wang, Tengfei Shi, Qiang Fu, Wei Yang, L. Huang, and Wei Liu. Towards playing full MOBA games with deep reinforcement learning. In *NeurIPS*, 2020.

Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre M. Bayen, and Yi Wu. The surprising effectiveness of MAPPO in cooperative, multi-agent games. *ArXiv*, abs/2103.01955, 2021.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. In *NeurIPS*, 2020.

Matthieu Zimmer, Claire Glanois, Umer Siddique, and Paul Weng. Learning fair policies in decentralized cooperative multi-agent reinforcement learning. In *ICML*, 2021.