A Game-Theoretic Approach to Multi-Agent Trust Region Optimization

Ying Wen[1,*], **Hui Chen**[2], **Yaodong Yang**[2], **Zheng Tian**[2], **Minne Li**[2], **Xu Chen**[3] **and Jun Wang**[2]
[1]Shanghai Jiao Tong University, [2]University College London, [3]Renmin University.

Abstract

Trust region methods are widely applied in single-agent reinforcement learning problems due to their monotonic performance-improvement guarantee at every iteration. Nonetheless, when applied in multi-agent settings, the guarantee of trust region methods no longer holds because an agent's payoff is also affected by other agents' adaptive behaviors. To tackle this problem, we conduct a game-theoretical analysis in the policy space, and propose a multi-agent trust region learning method (MATRL), which enables trust region optimization for multi-agent learning. Specifically, MATRL finds a stable improvement direction that is guided by the solution concept of Nash equilibrium at the meta-game level. We derive the monotonic improvement guarantee in multi-agent settings and show the local convergence of MATRL to stable fixed points in differential games. To test our method, we evaluate MATRL in both discrete and continuous multiplayer general-sum games including checker and switch grid worlds, multi-agent MuJoCo, and Atari games. Results suggest that MATRL significantly outperforms strong multi-agent reinforcement learning baselines.

1 Introduction

Multi-agent systems (MASs) [1] have received much attention from the reinforcement learning community [2]. In the real world, automated driving [3, 4], StarCraft II [5, 6] and Dota 2 [7] are a few examples of the myriad of applications that can be modeled by MASs. Due to the complexity of multi-agent problems [8], an investigation into whether agents can learn to behave effectively during interactions with environments and other agents is essential [9]. This investigation can be conducted naively through an *independent learner* (IL) [10], which ignores the other agents and optimizes the policy assuming a stable environment [11, 12]; and *trust region* method (e.g., proximal policy optimization (PPO) [13]) based ILs are popular [5, 7] due to their theoretical guarantee for single-agent learning [14] and good empirical performance in real-world applications.

In multi-agent scenarios, however, an agent's improvement is affected by other agents' adaptive behaviors (i.e., the multi-agent environment is *nonstationary* [12]). As a result, trust region learners can measure the policy improvements of agents' predicted policies compared to the current policies, but the improvements compared to the other agents' predicted policies are still unknown (shown in Fig. 1). Therefore, trust-region-based ILs perform worse in MASs than in single-agent tasks. Moreover, the convergence to a *fixed point*, such as a *Nash equilibrium* [15, 16, 17], is a common and widely accepted solution concept for multi-agent learning. Thus, although ILs can best respond to other agents' current policies, they lose their convergence guarantee [18].

One solution for addressing the convergence problem for ILs is empirical game-theoretic analysis (EGTA) [19], which approximates the best response to the policies generated by ILs [20, 21]. Although EGTA-based methods [20, 22, 23] establish convergence guarantees in several game

*Correspondence to Ying Wen <ying.wen@sjtu.edu.cn>.

Preprint. Under review.

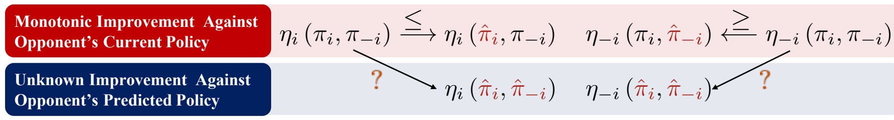


Figure 1: Discounted returns η_i for an agent i given different joint policy pairs, where π_i is the current policy, and π'_i is the simultaneously predicted policy. Given π_i, the monotonic improvements of a fixed opponent can be easily measured: $\eta_i(\pi'_i, \pi_{-i}) \geq \eta_i(\pi_i, \pi_{-i})$. However, due to simultaneous learning, the improvement of $\eta_i(\pi'_i, \pi'_{-i})$ is unknown compared to $\eta_i(\pi_i, \pi_{-i})$.

classes, their computational cost is also large when empirically approximating and solving the meta-game [24]. Other multi-agent learning approaches collect or approximate additional information such as communication [25, 6] and centralized joint critics [26, 27, 28, 29]. Nevertheless, these methods usually require centralized critics or centralized communication assumptions, which require extra training efforts. Thus, there is considerable interest in the use of multi-agent learning to find an algorithm that while having minimal requirements and computational cost as ILs, also simultaneously improves convergence performance.

This paper presents a *multi-agent trust region learning* (MATRL) algorithm that augments the trust region ILs with a meta-game analysis to improve learning stability and efficiency. In MATRL, a trust region trial step for an agent's payoff improvement is implemented by ILs, which provide a predicted policy based on the current policy. Then, an empirical policy-space meta-game is constructed to compare the expected advantages of the predicted policies with those of the current policies. By solving the meta-game, MATRL finds a restricted step by aggregating the current and predicted policies using the meta-game Nash equilibrium. Finally, MATRL takes the best responses based on the aggregated policies from the last step for each agent to explore because the identified stable trust region is not always strictly stable. MATRL is, therefore, able to provide a weakly stable solution compared to naive ILs. Based on a trust region IL, MATRL requires the knowledge of other agents' policy during the meta-game analysis but does not need extra centralized parameters, simulations, or modifications to the IL itself. We provide insights into the empirical meta-game in Section 2.2, showing that the approximated Nash equilibrium of the meta-game is a weak stable fixed point of the underlying game. Our experiments demonstrate that MATRL significantly outperforms deep ILs [13] with the same hyperparameters, VDN [30], QMIX [29] and QDPP [28] methods in discrete action grid worlds, decentralized PPO ILs, centralized MADDPG [26] and independent DDPG and COMIX [31] in a continuous action multi-agent MuJoCo task [31] and zero-sum multi-agent Atari [32].

2 Multi-Agent Trust Region Learning

Notations & Preliminaries. A stochastic game [33, 34] can be defined as follows: $\mathcal{G} = \langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}_i\}, \{\mathcal{R}_i\}, \mathcal{P}, p_0, \gamma \rangle$, where $\mathcal{N}$ is a set of agents, $n = |\mathcal{N}|$ is the number of agents, and $\mathcal{S}$ denotes the state space. $\mathcal{A}_i$ is the action space for agent i. $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n = \mathcal{A}_i \times \mathcal{A}_{-i}$ is the joint action space, and for simplicity, we use $-i$ to denote agents other than agent i. $\mathcal{R}_i = R_i(s, a_i, a_{-i})$ is the reward function for agent $i \in \mathcal{N}$. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the transition function. p_0 is the initial state distribution, and $\gamma \in [0, 1)$ is a discount factor. Each agent $i \in \mathcal{N}$ has a stochastic policy $\pi_i(a_i|s) : \mathcal{S} \times \mathcal{A}_i \rightarrow [0, 1]$ and aims to maximize its long-term discounted return:

$$\eta_i(\pi_i, \pi_{-i}) = \mathbb{E}_{s^0, a_i^0, a_{-i}^0 \cdots} \left[\sum_{t=0}^{\infty} \gamma^t R_i(s^t, a_i^t, a_{-i}^t) \right], \tag{1}$$

where $s^0 \sim p_0$, $s^{t+1} \sim \mathcal{P}(s^{t+1}|s^t, a_i^t, a_{-i}^t)$, and $a_i^t \sim \pi_i(a_i^t|\tau_i^t)$. Then, we have the standard definitions of the state-action value and state value functions: $Q_i^{\pi_i, \pi_{-i}}(s^t, a_i^t, a_{-i}^t) = \mathbb{E}_{s^{t+1}, a_i^{t+1}, a_{-i}^{t+1} \cdots}[\sum_{l=0}^{\infty} \gamma^l R_i(s^{t+l}, a_i^{t+l}, a_{-i}^{t+l})]$ and $V_i^{\pi_i, \pi_{-i}}(s^t) = \mathbb{E}_{a_i^t, a_{-i}^t, s^{t+1} \cdots}[\sum_{l=0}^{\infty} \gamma^l R_i(s^{t+l}, a_i^{t+l}, a_{-i}^{t+l})]$; also the advantage function $A_i^{\pi_i, \pi_{-i}}(s^t, a_i^t, a_{-i}^t) = Q_i^{\pi_i, \pi_{-i}}(s^t, a_i^t, a_{-i}^t) - V_i^{\pi_i, \pi_{-i}}(s^t)$, given the state and joint action.

Motivations. A trust region algorithm aims to answer two questions: how to compute a trial step and whether the trial step should be accepted. In multi-agent learning, a trial step toward agents payoff improvement can be easily implemented with ILs, denoted as ***independent improvement direction (IID)***. The remaining issue is resolved by finding a restricted step leading to a stable improvement

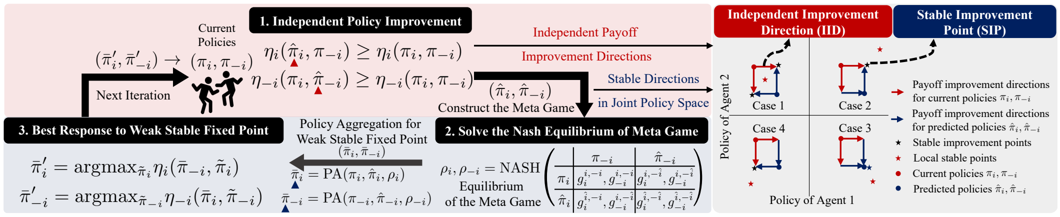


Figure 2: **(Left)**: Overview of the MATRL phases. The pale red area indicates independent payoff improvement directions; the pale blue area shows stable improvement directions in joint policy space and π: current policy, $\hat{\pi}$: predicted policy in IID step, $\bar{\pi}$: aggregated policy in SIP step; π', next policy. **(Right)**: the gray area illustrates IID and SIP with a two-agent game, in which the arrows indicate the payoff improvement directions for agents. The IID guarantees the partially monotone game in red arrows; then, the SIPs are determined by improvement directions (include four cases) of predicted policies in blue arrows.

direction, which is not in the single agent's policy space but in the joint policy space. In other words, MATRL decomposes trust region learning into two parts: first, an IID between *current policy* π_i and *predicted policy* $\hat{\pi}_i$ should be identified; then, with the help of the predicted policy, a more refined method, to some extent, can approximate a stable trial step. Instead of line searching in a single-agent payoff improvement [35] direction, MATRL searches for a joint policy space to achieve a conservative and stable improvement. Essentially, MATRL is an extension of the single-agent TRPO to a MAS, which learns to find a stable point between the current policy and the predicted policy. To find the stable improvement directions, we assume knowledge about other agents' policies during training to avoid unstable improvement via empirical meta-game analysis, while the execution can still be fully decentralized. We explain every step of MATRL in detail in the following sections (also in Fig. 2).

2.1 Independent Trust Payoff Improvement

Single-agent reinforcement learning algorithms can be straightforwardly applied to multi-agent learning, where we assume that all agents behave independently [10]. In this section, we have chosen the policy-based reinforcement learning method—ILs. In multi-agent games, the environment becomes a Markov decision process for agent i when each of the other agents plays according to a fixed policy. We set agent i to make a monotonic improvement against its opponents' fixed policies. Thus, at each iteration, the policy is updated by maximizing the utility function η_i over a local neighborhood of the current joint policy π_i, π_{-i}. We can adopt TRPO (or, PPO [13]), which constrains the step size in the policy update:

$$\hat{\pi}_i = \arg\max_{\pi \in \Pi_{\theta_i}} \eta_i(\pi, \pi_{-i}) \quad \text{s.t. } D\left(\pi_i, \hat{\pi}_i\right) \leq \delta_i, \tag{2}$$

where D is a distance measurement, and δ_i is a constant. Independent trust region learners produce the monotonically improved policy $\hat{\pi}_i$, which guarantees $\eta_i\left(\hat{\pi}_i, \pi_{-i}\right) \geq \eta_i\left(\pi_i, \pi_{-i}\right)$ and provides a trust payoff bound by $\hat{\pi}_i$. Due to simultaneous policy improvement without awareness of other agents , however, the lower bound of payoff improvement from single-agent [35] no longer holds for multi-agent payoff improvement. By following a similar logic in proof, we can obtain a precise lower bound for a simultaneous-move multi-agent payoff improvement.

Remark 1. *The approximated expected advantage $g_i^{\pi_i,\pi_{-i}}$ gained by agent i when $\pi_i, \pi_{-i} \to \hat{\pi}_i, \hat{\pi}_{-i}$ is denoted as follows:*

$$g_i^{\pi_i,\pi_{-i}}(\hat{\pi}_i, \hat{\pi}_{-i}) := \sum_s p^{\pi_i,\pi_{-i}}(s) \sum_{a_i,a_{-i}} \hat{\pi}_i(a_i|s)\hat{\pi}_{-i}(a_{-i}|s)A_i^{\pi_i,\pi_{-i}}(s, a_i, a_{-i}), \tag{3}$$

where $p^{\pi_i,\pi_{-i}}(s)$ discounted state visitation frequencies induced by π_i, π_{-i}. Then, the following lower bound can be derived for multi-agent independent trust region optimization:

$$\eta_i(\hat{\pi}_i, \hat{\pi}_{-i}) - \eta_i(\pi_i, \pi_{-i}) \geq g_i^{\pi_i,\pi_{-i}}(\hat{\pi}_i, \hat{\pi}_{-i}) - \frac{4\gamma\epsilon_i}{(1-\gamma)^2}(\alpha_i + \alpha_{-i} - \alpha_i\alpha_{-i})^2, \tag{4}$$

where $\epsilon_i = \max_{s,a_{-i},a_{-i}} \left|A_i^{\pi_i,\pi_{-i}}(s, a_i, a_{-i})\right|$, $\alpha_i = D_{\text{TV}}^{\max}\left(\pi_i, \hat{\pi}_i\right) = \max_s D_{\text{TV}}(\pi_i(\cdot|s)\|\hat{\pi}_i(\cdot|s))$ for agent i, and D_{TV} is the total variation divergence [35]. More details are included in Appendix B.

Based on the independent trust payoff improvement, although the predicted policy $\hat{\pi}_i$ will guide us in determining the step size of the IID, the stability of $(\hat{\pi}_i, \hat{\pi}_{-i})$ is still unknown. As shown in Remark 1, an agent's lower bound is approximately $O(4\alpha^2)$, which is four times larger than the single-agent lower bound trust region of $O(\alpha^2)$ [14]. Furthermore, $\epsilon_i = \max_{s,a_{-i},a_{-i}} \left|A_i^{\pi_i,\pi_{-i}}(s, a_i, a_{-i})\right|$ depends on other agents' action a_{-i}, which will be very large when agents have conflicting interests. Therefore, the most critical issue underlying MATRL is finding a Stable Improvement Point (SIP) after the IID. In the next section, we illustrate how to search for a weak stable fixed point within the IID based on the meta-game analysis.

2.2 Approximating the Weak Stable Fixed Point

Stabilizing the independent trust payoff improvements is one of the essential components of MATRL. Since each iteration of MATRL requires the solving of additional stable improvement subproblem, finding an efficient solver for this subproblem is very important. Instead of using the stable fixed points [36] as the stable improvement target, we choose the *weak stable fixed point* in Definition 2, which is much easier to find. To maximize the objective defined in Eq. (1), we can ask that *reasonable* algorithms avoid all strict minimums (unstable fixed points), which imposes only that agents are well-behaved regarding strict minima, even if their individual behaviors are not self-interested [37]. Before providing the clear definitions for these points, we first define a differentiable game restricted by the IID:

Definition 1 (Differentiable Restricted Game (DRG))**.** *If the policy space for each agent i in a game is restricted to open sets $\bar{\Pi}_i = [\pi_i, \hat{\pi}_i] \subseteq \Pi_i$, where $\bar{\Pi}_i \subseteq \Pi_i$, and the expected advantage g_i is twice continuously differentiable in this range, then we call it a differentiable restricted game.*

Denote the simultaneous gradient of the DRG as $\boldsymbol{\xi}(\pi_i, \pi_{-i}) = (\nabla_{\pi_i} g_i, \nabla_{\pi_{-i}} g_{-i})$. Adapted from [36] and [37], we introduce the Hessian of DRG as the block matrix $H = \nabla_{\pi_i,\pi_{-i}} \boldsymbol{\xi}(\pi_i, \pi_{-i})$ to define the types of fixed points:

Definition 2 (Weak Stable Fixed Point)**.** *A point $(\bar{\pi}_i, \bar{\pi}_{-i})$ is a fixed point if $\boldsymbol{\xi}(\bar{\pi}_i, \bar{\pi}_{-i}) = \mathbf{0}$. We then say that $(\bar{\pi}_i, \bar{\pi}_{-i})$ is stable if $H(\bar{\pi}_i, \bar{\pi}_{-i}) \preceq 0$, is unstable if $H(\bar{\pi}_i, \bar{\pi}_{-i}) \succ 0$ and is a* ***weak stable fixed point*** *if $H(\bar{\pi}_i, \bar{\pi}_{-i}) \not\succ 0$*[2].

We denote the weak stable fixed points in the DRG as the ***stable improvement point (SIP)***, it is reasonable if it converges only to fixed points and avoids unstable fixed points (strict minimum) almost completely. Given that we already have the IID, which produces a predicted policy, with the knowledge about all agents policies, it is natural to conduct an EGTA [38] to search for a SIP in the area bounded by the current and predicted policy pair. We then define a meta-game in which each agent i has only two strategies $\pi_i, \hat{\pi}_i$:

$$\mathcal{M}(\pi_i, \hat{\pi}_i, \pi_{-i}, \hat{\pi}_{-i}) = \begin{pmatrix} g_i^{i,-i}, g_{-i}^{i,-i} & g_i^{i,-\hat{i}}, g_{-i}^{i,-\hat{i}} \\ g_i^{\hat{i},-i}, g_{-i}^{\hat{i},-i} & g_i^{\hat{i},-\hat{i}}, g_{-i}^{\hat{i},-\hat{i}} \end{pmatrix}, \tag{5}$$

where $g_i^{\hat{i},-\hat{i}} = g_i^{\pi_i,\pi_{-i}}(\hat{\pi}_i, \hat{\pi}_{-i})$ (as defined in Eq. (3)) is an empirical payoff entry of the meta-game, and note that $g_i^{i,-i} = 0$, as it has an expected advantage over itself. Compared with using $\eta_i(\hat{\pi}_i, \hat{\pi}_{-i}) = \eta_i(\pi_i, \pi_{-i}) + g_i^{\hat{i},-\hat{i}}$ as the meta-game payoff, $g_i^{\hat{i},-\hat{i}}$ has lower variance and is easier to approximate because $\eta_i(\pi_i, \pi_{-i})$ is a constant baseline. However, most entries in $\mathcal{M}$ are unknown, and many extra simulations are required to estimate the payoff entries (e.g., $g_i^{\hat{i},-\hat{i}}$) in EGTA. Instead, we reuse the trajectories in the IID step to approximate $g_i^{\hat{i},-\hat{i}}$ by ignoring the small changes in the state visitation density caused by $\pi_i \to \hat{\pi}_i$.

Remark 2. *The meta-game $\mathcal{M}(\pi_i, \hat{\pi}_i, \pi_{-i}, \hat{\pi}_{-i})$ is a partially monotone game and has a pure strategy equilibrium [39], because the monotonic improvements $g_i^{i,-i} \leq g_i^{\hat{i},-i}$ and $g_{-i}^{i,-i} \leq g_{-i}^{i,-\hat{i}}$ when $\pi_i, \pi_{-i} \to \hat{\pi}_i, \hat{\pi}_{-i}$.*

Taking the two-agent case as an example, as we can see in Eq. (5), meta-game $\mathcal{M}$ becomes a 2×2 matrix-form game, which is much smaller in size than the whole underlying game. Besides, according to Fig. 2 Right and Remark 2, all four cases have at least one pure strategy that leads a

[2] In this paper, we want to maximize the return, not minimize the loss, so we need to avoid a strict minimum.

stable improvement direction. To this end, we can use the existing Nash solvers (e.g.,CMA-ES [40]) for matrix-form games to compute a Nash equilibrium $\rho_i, \rho_{-i} = \mathrm{NashSolver}(\mathcal{M})$ for meta-game $\mathcal{M}$, where ρ_i and $\rho_{-i} \in [0, 1]$, and the Nash equilibrium of the meta-game is also an approximated equilibrium of the restricted underlying game [41]. Then, SIP policies $\bar{\pi}_i, \bar{\pi}_{-i}$ can be aggregated based on current policy π_i and predicted policy $\hat{\pi}_i$ in the IID for each agent i.

Assumption 1. *In the IID step, ILs enjoy the monotonic improvement against fixed opponent policies, in which the change from π_i to $\hat{\pi}_i$ is usually constrained by a small step size. Therefore, it is reasonable to assume that there is a linear, continuous and monotonic change in the restricted policy space between π_i and $\hat{\pi}_i$.*

In this case, with ρ_i being agent i's Nash equilibrium policy in the meta-game, $\bar{\pi}_i$ can be derived via a linear mixture: $\bar{\pi}_i = \rho_i \pi_i + (1 - \rho_i)\hat{\pi}_i$, which delimits agent i's SIP. Now, we can prove that $(\bar{\pi}_i, \bar{\pi}_{-i})$ is a weak stable fixed point for the underlying game in Theorem 1. Furthermore, based on Assumption 1, the payoff and policy space $[\pi_i, \hat{\pi}_i]$ for DRG are bounded in a linear continuous space, we can conclude the following theorem:

Theorem 1 (Existence of a Weak Stable Fixed Point)**.** *If (ρ_i, ρ_{-i}) is a Nash equilibrium of the meta-game $\mathcal{M}$, then linear mixture joint policy $(\bar{\pi}_i, \bar{\pi}_{-i})$ is a weak stable fixed point for the DRG.*

Proof. See Appendix C. □

According to Theorem 1, $(\bar{\pi}_i, \bar{\pi}_{-i})$ is a weak stable fixed point of the restricted underlying game. Although the weak stable fixed point is relatively weak compared to the stable fixed points [36], as we have stated, a weak stable fixed point is a reasonable (not as strong as it is rational) requirement for an algorithm to avoid the minimum. Furthermore, weak stable fixed points can suit general game settings. As shown in Appendix C, in cooperative, competitive, and general-sum games, the fixed points found by the meta-game analysis can be either stable or saddle points. Similarly, a local Nash equilibrium can be stable or saddle in different games [17]. Therefore, the goodness of stable concepts depends on specific settings. If we make some additional game class assumptions, then we can easily obtain stronger fixed point types. Nevertheless, this approach comes with a cost, requiring additional computation or assumptions that may break the most general settings. In addition, when the meta-game has multiple Nash equilibria, an equilibrium is randomly selected in our work.

2.3 Improvement over a Weak Stable Fixed Point

Although the weak stable fixed point, $(\bar{\pi}_i, \bar{\pi}_{-i})$, binds the policy update to another fixed point, there are still fully stable points according to Theorem 1. Besides, it is difficult to generalize for the other parts of the policy space not reached by SIP, especially in anticoordination games [20]. Similar to the extragradient method [42], to encourage the exploration, we apply the best response against the weak stable fixed point $(\bar{\pi}_i, \bar{\pi}_{-i})$:

$$\pi_i' = \arg\max_{\pi \in \Pi_{\theta_i}} \eta_i\left(\pi, \bar{\pi}_{-i}\right). \tag{6}$$

Algorithm 1 MATRL

Input: Initialize policies π_i for each i.

for $k \in \{0, 1, 2, \cdots\}$ **do**
  Use current policies π_i, π_{-i} to collect trajectories.
  for each i **do**
    Compute a one-step predicted policy $\hat{\pi}_i$ (Eq. (2)).
  end for
  Solve meta-game $\mathcal{M}(\pi_i, \hat{\pi}_i, \pi_{-i}, \hat{\pi}_{-i})$.
  Compute weak stable fixed point $\bar{\pi}_i, \bar{\pi}_{-i}$.
  For each i: Compute best response π_i' (Eq. (6)).
  $\pi_i \leftarrow \pi_i'$, $\pi_{-i} \leftarrow \pi'_{-i}$.
end for

To perform the best response, we need another round to collect the experiences and perform a gradient step in Eq. (6). However, in practice, since we already have the trajectories in the IID step, the best response to the weak stable fixed point can be easily estimated through importance sampling. Alternatively, by defining $c_i \overset{\text{def}}{=} \min\left(1 + \bar{c}, \max(1 - \bar{c}, \frac{\pi_i(a_i|s)}{\bar{\pi}_i(a_i|s)})\right)$ as truncated importance sampling weights, we can rewrite the best response update to Eq. (6) as an equivalent form to the following one in terms of expectations: $\pi_i' = \arg\max_\pi \mathbb{E}_{a_{-i} \sim \bar{\pi}_{-i}}[c_{-i}\eta_i\left(\pi_i, \pi_{-i}\right)]$. If the agents end up playing the BR, then there is no further improvement in the IID step; the payoff entries in the restricted meta-game would be zero, meaning agents will stay at the current policies following MATRL steps.

2.4 Local Convergence

MATRL is a gradient-based algorithm with the best response to policies within the SIPs, which is essentially a variant of LookAhead methods [43, 44, 45]. More specifically, MATRL enhances the classic LookAhead method with variable step size scaling [46] or two time-scale update rules [47] at each SIP step, which is controlled by restricted meta-game analysis. It has been proven that the LookAhead method can locally converge to a stable fixed point and avoid strict saddles in all differentiable games [45, 48, 49]. Similarly, we show the local convergence of MATRL in Theorem 2. Please note, here, that to investigate the convergence, fixed point iterations are conducted on the whole learning process, while the meta-game analysis step in MATRL borrows the variable stepsize scaling and shows it is reasonable to locally avoid unstable fixed points. Unlike LOLA, which uses a first-order Taylor expansion to estimate the best response to a predicted policy, we elaborately design the look-ahead step within the SIPs and perform the gradient steps for the best response to the SIPs. We also show that MATRL empirically outperforms the typical LookAhead method, IL LookAhead (IL-LA), in the experiments. As shown in Fig. 3, we compare the convergence of gradient decent IL, Extragradient, LookAhead and MATRL in toy differential game[3] with strong rotational force, where MATRL has faster convergence to the stable fixed point.

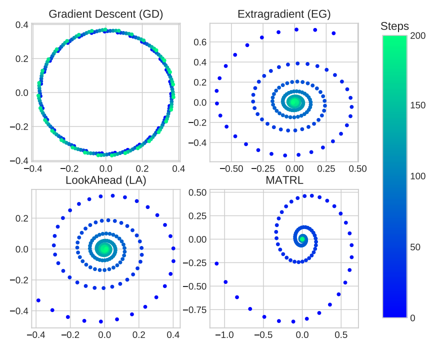


Figure 3: Learning the dynamics of MATRL in a rotational differential game. GD cannot converge; EG, LA and MATRL converge to the stable fixed point and MATRL has the fastest convergence speed with same learning rate 0.02.

Theorem 2 (Local Convergence of MATRL)**.** *Let the objectives $\eta_i(\pi_i, \pi_{-i})$ of agents are twice continuously differentiable and step size α is sufficiently small, MATRL converges locally to a stable fixed point with ϵ error in Euclidean distance.*

Proof. See Appendix D. □

2.5 Discussions

Computation Cost. Compared to pure ILs, there are two extra cost sources in common meta-game analysis: approximating and solving the meta-game [21]. In our case, the meta-game is restricted to a local two-action game, where two actions, π_i and $\hat{\pi}_i$, are close to each other. Reusing the IID trajectories will some estimation errors [41], but this issue can be eased by large batch size. Then, we can enjoy this proximity property and reduce the meta-game approximation cost (without extra sampling) by reusing the collected trajectories in the IID step. The next crucial problem is how to solve the n-agent two-action meta-game, which consists of the 2^n entries of each of the n payoff matrices. Solving this meta-game is much simpler than solving the whole underlying game, which increases exponentially with state size, action size, agent number, and time horizons. As the general-sum matrix-form game has no fully polynomial time approximation for computing Nash equilibria [50], it usually costs a great deal to solve the game [51]. However, as shown in Remark 2, there always exists at least one pure Nash equilibrium in the meta-game, which can be computed in polynomial time [52]. Therefore, if we only require an approximated Nash equilibrium, then when n is small, for example, $n \leq 5$, it is affordable to find a meta-game Nash equilibrium with subexponential complexity [53]. But this problem still exists when n is large. In this case, we can try a mean field approximation [54] or utilize special payoff structure assumptions (e.g., graphical game [55, 51]) in the meta-game to reduce computational complexity.

Connections to Existing Methods. MATRL generalizes many existing methods with the best response. In extreme cases, where the meta-game Nash equilibrium is $(\rho_i, \rho_{-i}) = (1, 1)$, which means that the Nash aggregated policies always maintain the current policies, MATRL degenerates to ILs. Here, we always best respond to other agents' current policy π_i and $\pi'_i = \arg\max_{\pi_i} \eta_i(\pi_i, \pi_{-i})$ following Eq. (6). The LookAhead [43, 44, 45], extragradient [56] and exploitability descent [57, 58] methods are also special instances of MATRL when meta-game Nash is $(\rho_i, \rho_{-i}) = (0, 0)$, which means that the best response to the most aggressive predicted policy $\hat{\pi}_{-i}$ and $\pi'_i = \arg\max_{\pi_i} \eta_i(\pi_i, \hat{\pi}_{-i})$. More specifically, let ξ denotes the game's simultaneous gradient,

[3] A two-agent differential game adopted from [36]. The loss functions: $\eta_i(\pi_i, \pi_{-i}) = \frac{1}{2}\pi_i^2 + 10\pi_i\pi_{-i}, \eta_{-i}(\pi_i, \pi_{-i}) = \frac{1}{2}\pi_{-i}^2 - 10\pi_i\pi_{-i}$.

H_o is the matrix of anti-diagonal blocks of H (Hessian of the game), and α is step-size. Then we can have the updating gradient for LookAhead methods as $(I - \alpha H_o)\,\xi$. Similarly, for MATRL, we have the updating gradient $(I - \rho\alpha H_o)\,\xi$, where ρ is a ratio determined by meta-game Nash to dynamically adjust the step-size at each iteration.

In summary, independent trust region learners' learning in MATRL will be constrained by a weak stable fixed point. By analyzing the relatively simpler meta-game, we can easily approximate this weak stable fixed point without extra rollouts or simulation. Although MATRL's training is centralized, its execution is fully decentralized, and it also does not require any extra centralized parameters or higher-order gradient computation. Fig. 2 presents an overview of MATRL. We also give the pseudocode of MATRL in Algo. 1, which is compatible with any policy-based IL.

3 Related Work

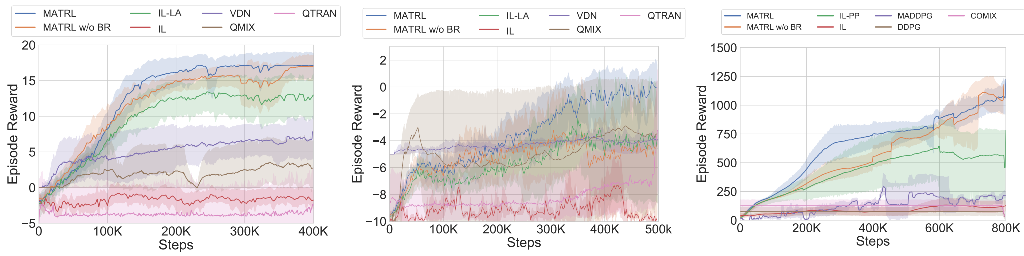
The study of gradient-based methods in multi-agent learning is quite extensive [17, 11]. Some works on learning in games have mostly focused on adjusting the step size, which attempts to use a multitimescale learning scheme [59, 60, 46] to achieve convergence. [36, 61, 45] tried to utilize second-order methods to shape the step size. However, the computational cost for second-order methods is very limiting in many cases. Other approaches include recursive reasoning techniques [62, 63] where agents explicitly take into account how their behaviors are going to affect their opponents during the gradient updates. Alternatively, MATRL approximates the second-order fixed-point information via a small meta-game with less cost compared to real Hessian computation. An alternative augments the gradient-based algorithms with the best response to the predicted polices [56, 43, 64, 44, 57, 58], which targets the challenge of instability caused by agents' change policies. Instead of taking the best response to the approximated opponent's policy, MATRL exploits the ideas from both streams and introduces an improvement over the weak stable fixed point.

The research also focuses on the EGTA [38, 65, 41], which creates a policy-space meta-game for modeling multi-agent interactions. Using various evaluation metrics, this work then updates and extends the policies based on the analysis of meta policies [20, 21, 22, 23, 24]. Although these methods are broad with respect to multi-agent tasks, they require extensive computing resources to estimate the empirical meta-game and solve it with its increasing size [22, 24]. In our method, we adopt the idea of a policy-space meta-game to approximate the fixed point. Unlike previous works, we only maintain current and predicted policies to construct the meta-game, which is computationally achievable in most cases. The payoff entry in MATRL's meta-game is the expected advantage, which has a lower estimation variance compared to the commonly used empirically estimated return in EGTAs. Regardless, we can reuse the trajectories in the IID step to estimate the payoffs without incurring additional sampling costs.

Recently, due to the use of neural networks as a function approximation for policies and values, many works have emerged on deep reinforcement learning (DRL) [66, 67]. TRPO [14, 35, 13] is one of the most successful DRL methods in the single-agent setting, which places constraints on the step size of policy updates, monotonically preserving any improvements. Based on the monotonic improvement in single-agent TRPO [35], MATRL extends the improvement guarantee to the multi-agent level towards a weak stable fixed point. Some works have directly applied fully decentralized single-agent DRL methods [10], which can be unstable during learning due to the issue of nonstationarity. However, [25, 68, 69] added an extra communication channel during the training and execution in a centralized way to avoid this nonstationarity issue. [30, 29, 27, 26] further exploit the setting of centralized learning decentralized execution (CTDE). These methods provide solutions for training agents in complex multi-agent environments, and the experimental results show their effectiveness compared with ILs. Similar to the CTDE setting, MATRL also enjoys fully decentralized execution. Although MATRL still needs knowledge about other agents' policies in adjusting the step size during training, it does not need centralized critics or any communication channels. Besides, [70, 71] attempted to apply trust-region methods in networked multi-agent settings by conducting consensus optimization with their neighbors. Instead takes a game-theoretical approach to compute the meta-game Nash to find policy improvement directions without networked assumption.

4 Experiments

We design experiments to answer the following questions: 1) Can the MATRL method empirically contribute to convergence in general game settings, including cooperative/competitive and



(a) Two-agent checker. (b) Four-agent switch. (c) Three-agent MuJoCo hopper.

Figure 4: Learning curves in discrete and continuous tasks. The solid lines are average episode returns with 10 random seeds for each model, and the light color areas are the error bars.

continuous/discrete games? 2) How is the performance of MATRL compared to ILs with the same hyperparameters and other strong MARL baselines in discrete and continuous games with various agent numbers? 3) Do the meta-game and best response to the weak stable fixed point bring about benefits? We first evaluate the convergence performance of MATRL in matrix form games to answer the first question and validate the effectiveness of convergence. For Question 2, we show that MATRL largely outperforms ILs (PPO [13]) and other centralized baselines (QMIX [29], QTRAN [72] and VDN [30]) in discrete grid world games that have coordination problems. MATRL also outperforms DDPG [67], MADDPG [26] and COMIX [31] for continuous multi-agent MuJoCo games. In addition, we test the algorithms with a 2-agent Atari Pong game to investigate whether MATRL can mitigate unstable cyclic behaviors [23] in zero-sum games. In these tasks, MATRL uses the same PPO configurations as ILs to examine the effectiveness of the trust region gradient-update mechanism, and we use official implementations for the other baselines. The step-by-step PPO-based MATRL algorithm is given in Appendix A. Finally, ablation studies are conducted by: 1. removing the best response, called the "MATRL w/o BR"; 2. skipping the SIP estimation, named "IL-LA", which has similar procedures as those of LOLA [44, 43], which approximates the best response to the predicted policies via Taylor expansion, but IL-LA takes the best response gradient steps for the predicted policies. These configurations provide insights into how much, if at all, the SIP and the best response contribute to the MATRL's performance. We also provide more environmental details and extra experimental results, in Appendices E and F, with detailed experimental settings and hyperparameters used for the algorithms. The code and experiment scripts are also anonymously available at `https://github.com/matrl-project/matrl`.

Table 1: Convergence rate and average convergence step in $1,000$ random 2×2 matrix games. MATRL shows slightly better convergence rate and speed compared to IGA-LA.

	Convergence Rate (in %) / Average Convergence Step		
Algo.	Coordination	Anticoordination	Cyclic
IGA	**99** $\pm$ 0.1 / 140.67 $\pm$ 105	97.5 $\pm$ 0.13 / 88.95 $\pm$ 130	78.0 $\pm$ 0.45 / 452.92 $\pm$ 202
IGA-LA	**99** $\pm$ 0.1 / 138.56 $\pm$ 105	97.5 $\pm$ 0.08 / 83.11 $\pm$ 129	80.9 $\pm$ 0.43 / 432.98 $\pm$ 206
MATRL	**99** $\pm$ 0.1 / **86.54** $\pm$ 77	**98.3** $\pm$ 0.08 / **75.52** $\pm$ 119	**84.6** $\pm$ 0.36 / **369.40** $\pm$ 200

Random 2×2 Matrix Games. To adequately examine MATRL in matrix games, we randomly generate three thousand 2×2 games of three types: coordination, anticoordination, and cyclic [73]. We choose the IGA and IGA-LA [43] as baselines and use IGA [74] as the ILs of MATRL. The results in Table 1 show that MATRL has a higher convergence rate, fewer steps for convergence and more stable performance in all types of games. More details about game generation and the effects of the learning dynamics are provided in Appendix E.

Grid Worlds. We evaluated MATRL in two grid world games from MA-Gym [75], two-agent checker, and four-agent switch, which are similar to the games in [30] but with more agents to examine if MATRL can handle the games that have more than two agents. In the checker game, two agents cooperate in collecting fruit on the map; the sensitive agent obtains 5 for an apple and -5 for a lemon, while the other agent obtains 1 and -1, respectively. Therefore, the optimal solution is to let the sensitive agent obtain the apple and the less sensitive agent obtain the lemon. In the four-agent switch game, two rooms are connected by a corridor, each room has two agents, and the four agents try to go through one corridor to the target in the opposite room. Only one agent can pass through the corridor at one time, and agents obtain -0.1 for each step and 5 for reaching the target, so they need to cooperate to obtain optimal scores. In both games, agents can move in four directions and only partially observe their position. Although our formulation uses a fully observable setting, in this game, the methods are adapted to the partially observable setting by pretending the observation

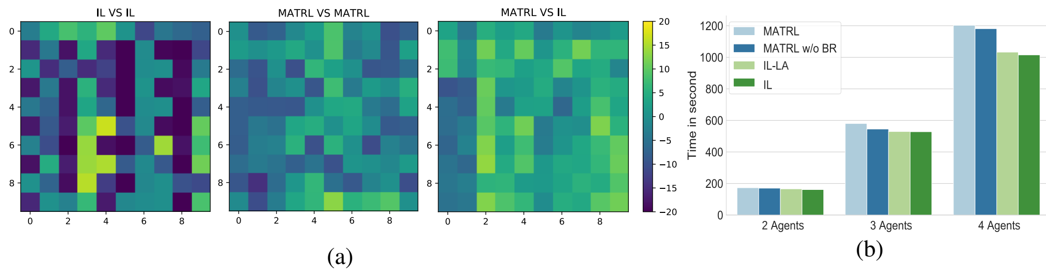


Figure 5: **(a)**: MATRL/IL versus MATRL/IL in the two-agent Pong game. For each setting, the grids show pairwise performance (average scores) by pitting their ten checkpoints against one another; yellow indicates a higher score. **(b)**: Run time for 20,000 environment steps (including 50 gradient steps) for the algorithms in two- to four-agent games.

is a state. We compare the MATRL with the PPO-based IL and two off-policy centralized training and decentralized execution baselines: VDN [30], QTRAN [72] and QMIX [29]. The results are given in Figs. 4a and 4b, where MATRL shows stable improvement and outperforms other baselines. In a two-agent checker game, using the best response, our method can achieve a total reward of 18, while the ILs' reward stays at -2. In addition, although PPO-based MATRL uses on-policy learning, it achieves better final results in fewer time steps compared to the off-policy baselines. For the four-agent switch game, as shown in Fig. 4b, MATRL can continuously improve the total rewards to 6.5, which is the closest to the optimal score for this game when compared with other baselines. The result of the four-agent switch also demonstrates the effectiveness of MATRL in guaranteeing stable policy improvement for games that have more than two agents.

Multi-Agent MuJoCo. We also examined MATRL in a multi-agent continuous control task with a three-agent hopper from [31]. Here, three agents cooperatively control each part of a hopper to move forward. The agents are rewarded with the distance traveled and the number of steps they make before falling. Fig. 4c shows that MATRL significantly outperforms ILs, MADDPG, DDPG, and the benchmarks like COMIX in [31] within the same amount of time. More results in multi-agent MuJoCo tasks (2-agent ant and 2-agent swimmer) are available in Appendix E.

Multi-Agent Atari Pong Game. In the 2-agent Pong game experiments, we used raw pixels as observations and trained the MATRL and IL agents independently. Following training, we compare the pairwise performance of these models by pitting their ten checkpoints against one another and recording average scores. We report the results in Fig. 5a, which shows that MATRL outperforms ILs in MATRL vs. IL settings in most policy pairs. In addition, from the MATRL vs. MATRL and ILs vs. IL settings' results, we can see that MATRL has a more transitive learning process than that of ILs, which means that MATRL can mitigate the common cyclic behaviors in zero-sum games.

Effect and Cost of the SIP and Best Response to a Fixed Point. This section analyzes the effect of the SIP from the meta-game Nash equilibrium and the best response against the weak stable fixed point. The ablation settings are obtained by removing the SIP (IL-LA) and the best response (MATRL w/o BR). In Fig. 4, we can observe that in all the tasks, without the best response to the fixed point, the learning curves of MATRL w/o BR have higher variance and the lowest final scores. This establishes the importance of the best response to stabilize and improve agents' performance and empirically shows that MATRL has better convergence ability than do the other baselines. Additionally, without the SIP to select a fixed point, MATRL recovers to ILs with policy prediction (IL-LA) [43, 44]. Similarly, the curves of IL-LA have lower final scores, and the convergence speed is not as good as that of MATRL, which suggests that the SIP provides benefits. MATRL w/o BR has lower variance compared to IL-LA, which reveals that the SIP can stabilize the learning via weak stable fixed point constraints. Finally, when compared to IL and IL-LA, as shown in Fig. 5b, in two- to four-agent games with 20,000 environment steps and 50 gradient steps, the training time of MATRL is empirically approximately 1.1-1.2 times slower. Given the significant performance improvement, we believe such extra computational cost from the SIP and the best response are acceptable.

5 Conclusions

We proposed and analyzed the trust region method for multi-agent learning problems, which considers the IID and SIP to meet multi-agent learning objectives. In practice, based on independent trust payoff learners, we provide a convenient way to approximate a further restricted step size within the SIP via a meta-game. This approach ensures that MATRL is generalized, flexible, and easily implemented to deal with multi-agent learning problems in general. Our experimental results justify the fact that the MATRL method significantly outperforms ILs using the same configurations and other strong MARL baselines in both continuous and discrete games with varying numbers of agents.

References

[1] Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

[2] Yaodong Yang and Jun Wang. An overview of multi-agent reinforcement learning from game theoretical perspective. *arXiv preprint arXiv:2011.00583*, 2020.

[3] Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics*, 9(1):427–438, 2012.

[4] Ming Zhou, Jun Luo, Julian Villela, Yaodong Yang, David Rusu, Jiayu Miao, Weinan Zhang, Montgomery Alban, Iman Fadakar, Zheng Chen, et al. Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving. *arXiv preprint arXiv:2010.09776*, 2020.

[5] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[6] Peng Peng, Ying Wen, Yaodong Yang, Quan Yuan, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2017.

[7] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[8] Krishnendu Chatterjee, Rupak Majumdar, and Marcin Jurdziński. On nash equilibria in stochastic games. In *International Workshop on Computer Science Logic*, pages 26–40. Springer, 2004.

[9] Drew Fudenberg, Fudenberg Drew, David K Levine, and David K Levine. *The theory of learning in games*, volume 2. MIT press, 1998.

[10] Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pages 330–337, 1993.

[11] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. Multi-agent reinforcement learning: An overview. In *Innovations in multi-agent systems and applications-1*, pages 183–221. Springer, 2010.

[12] 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.

[13] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[14] Sham M. Kakade and John Langford. Approximately optimal approximate reinforcement learning. In Claude Sammut and Achim G. Hoffmann, editors, *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), University of New South Wales, Sydney, Australia, July 8-12, 2002*, pages 267–274. Morgan Kaufmann, 2002.

[15] John F Nash. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

[16] Michael Bowling and Manuela Veloso. Existence of multiagent equilibria with limited agents. *Journal of Artificial Intelligence Research*, 22:353–384, 2004.

[17] Eric Mazumdar, Lillian J Ratliff, and S Shankar Sastry. On gradient-based learning in continuous games. *SIAM Journal on Mathematics of Data Science*, 2(1):103–131, 2020.

[18] Guillaume J Laurent, Laëtitia Matignon, Le Fort-Piat, et al. The world of independent learners is not markovian. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 15(1):55–64, 2011.

[19] Michael P Wellman. Methods for empirical game-theoretic analysis. In *AAAI*, pages 1552–1556, 2006.

[20] Marc Lanctot, Vinícius Flores Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4190–4203, 2017.

[21] Paul Muller, Shayegan Omidshafiei, Mark Rowland, Karl Tuyls, Julien Pérolat, Siqi Liu, Daniel Hennes, Luke Marris, Marc Lanctot, Edward Hughes, Zhe Wang, Guy Lever, Nicolas Heess, Thore Graepel, and Rémi Munos. A generalized training approach for multiagent learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[22] Shayegan Omidshafiei, Christos Papadimitriou, Georgios Piliouras, Karl Tuyls, Mark Rowland, Jean-Baptiste Lespiau, Wojciech M Czarnecki, Marc Lanctot, Julien Perolat, and Remi Munos. α-rank: Multi-agent evaluation by evolution. *Scientific reports*, 9(1):1–29, 2019.

[23] David Balduzzi, Marta Garnelo, Yoram Bachrach, Wojciech Czarnecki, Julien Pérolat, Max Jaderberg, and Thore Graepel. Open-ended learning in symmetric zero-sum games. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 434–443. PMLR, 2019.

[24] Yaodong Yang, Rasul Tutunov, Phu Sakulwongtana, and Haitham Bou Ammar. $\alpha\alpha$-rank: Practically scaling α-rank through stochastic optimisation. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1575–1583, 2020.

[25] Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2137–2145, 2016.

[26] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6379–6390, 2017.

[27] Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 2974–2982. AAAI Press, 2018.

[28] Yaodong Yang, Ying Wen, Jun Wang, Liheng Chen, Kun Shao, David Mguni, and Weinan Zhang. Multi-agent determinantal q-learning. In *International Conference on Machine Learning*, pages 10757–10766. PMLR, 2020.

[29] Tabish Rashid, Mikayel Samvelyan, Christian Schröder de Witt, Gregory Farquhar, Jakob N. Foerster, and Shimon Whiteson. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4292–4301. PMLR, 2018.

[30] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th international conference on autonomous agents and multiagent systems*, pages 2085–2087. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[31] Christian Schroeder de Witt, Bei Peng, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. Deep multi-agent reinforcement learning for decentralized continuous cooperative control, 2020.

[32] Justin K Terry and Benjamin Black. Multiplayer support for the arcade learning environment. *arXiv preprint arXiv:2009.09341*, 2020.

[33] Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.

[34] Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.

[35] John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1889–1897. JMLR.org, 2015.

[36] David Balduzzi, Sébastien Racanière, James Martens, Jakob N. Foerster, Karl Tuyls, and Thore Graepel. The mechanics of n-player differentiable games. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 363–372. PMLR, 2018.

[37] Alistair Letcher. On the impossibility of global convergence in multi-loss optimization. *arXiv preprint arXiv:2005.12649*, 2020.

[38] Karl Tuyls, Julien Perolat, Marc Lanctot, Joel Z Leibo, and Thore Graepel. A generalised method for empirical game theoretic analysis. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 77–85. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[39] Jun-ichi Takeshita and Hidefumi Kawasaki. Necessity and sufficiency for the existence of a pure-strategy nash equilibrium. *Journal of the Operations Research Society of Japan*, 55(3):192–198, 2012.

[40] Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*, 11(1):1–18, 2003.

[41] Karl Tuyls, Julien Perolat, Marc Lanctot, Edward Hughes, Richard Everett, Joel Z Leibo, Csaba Szepesvári, and Thore Graepel. Bounds and dynamics for empirical game theoretic analysis. *Autonomous Agents and Multi-Agent Systems*, 34(1):7, 2020.

[42] Panayotis Mertikopoulos, Bruno Lecouat, Houssam Zenati, Chuan-Sheng Foo, Vijay Chandrasekhar, and Georgios Piliouras. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[43] Chongjie Zhang and Victor R. Lesser. Multi-agent learning with policy prediction. In Maria Fox and David Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press, 2010.

[44] Jakob Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 122–130. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[45] Alistair Letcher, Jakob N. Foerster, David Balduzzi, Tim Rocktäschel, and Shimon Whiteson. Stable opponent shaping in differentiable games. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[46] Michael Bowling and Manuela Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136(2):215–250, 2002.

[47] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems*

30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 6626–6637, 2017.

[48] Constantinos Daskalakis, Dylan J Foster, and Noah Golowich. Independent policy gradient methods for competitive reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5527–5540. Curran Associates, Inc., 2020.

[49] Kaiqing Zhang, Zhuoran Yang, and Tamer Basar. Policy optimization provably converges to nash equilibria in zero-sum linear quadratic games. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[50] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Computing nash equilibria: Approximation and smoothed complexity. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 603–612. IEEE, 2006.

[51] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

[52] Alex Fabrikant, Christos Papadimitriou, and Kunal Talwar. The complexity of pure nash equilibria. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 604–612, 2004.

[53] Richard J Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *Proceedings of the 4th ACM conference on Electronic commerce*, pages 36–41, 2003.

[54] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5567–5576. PMLR, 2018.

[55] Michael L. Littman, Michael J. Kearns, and Satinder P. Singh. An efficient, exact algorithm for solving tree-structured graphical games. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 817–823. MIT Press, 2001.

[56] Anatoly Antipin. Extragradient approach to the solution of two person non-zero sum games. In *Optimization and Optimal Control*, pages 1–28. World Scientific, 2003.

[57] Jie Tang, Keiran Paster, , and Pieter Abbeel. Equilibrium finding via asymmetric self-play reinforcement learning. *Deep Reinforcement Learning Workshop NeurIPS 2018*, 2018.

[58] Edward Lockhart, Marc Lanctot, Julien Pérolat, Jean-Baptiste Lespiau, Dustin Morrill, Finbarr Timbers, and Karl Tuyls. Computing approximate equilibria in sequential adversarial games by exploitability descent. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 464–470. ijcai.org, 2019.

[59] David S Leslie and Edmund J Collins. Individual q-learning in normal form games. *SIAM Journal on Control and Optimization*, 44(2):495–514, 2005.

[60] David S Leslie, EJ Collins, et al. Convergent multiple-timescales reinforcement learning algorithms in normal form games. *The Annals of Applied Probability*, 13(4):1231–1251, 2003.

[61] Eric V Mazumdar, Michael I Jordan, and S Shankar Sastry. On finding local nash equilibria (and only local nash equilibria) in zero-sum games. *arXiv preprint arXiv:1901.00838*, 2019.

[62] Ying Wen, Yaodong Yang, Rui Luo, Jun Wang, and Wei Pan. Probabilistic recursive reasoning for multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2018.

[63] Ying Wen, Yaodong Yang, Rui Luo, and Jun Wang. Modelling bounded rationality in multi-agent interactions by generalized recursive reasoning. *arXiv preprint arXiv:1901.09216*, 2019.

[64] Tianyi Lin, Zhengyuan Zhou, Panayotis Mertikopoulos, and Michael I. Jordan. Finite-time last-iterate convergence for multi-agent learning in games. In *Proceedings of the 37th International*

Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of *Proceedings of Machine Learning Research*, pages 6161–6171. PMLR, 2020.

[65] Patrick R Jordan and Michael P Wellman. Generalization risk minimization in empirical game models. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 553–560, 2009.

[66] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[67] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[68] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2244–2252, 2016.

[69] Peng Peng, Ying Wen, Yaodong Yang, Quan Yuan, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2017.

[70] Wenhao Li, Xiangfeng Wang, Bo Jin, Junjie Sheng, and Hongyuan Zha. Dealing with non-stationarity in multi-agent reinforcement learning via trust region decomposition. *arXiv preprint arXiv:2102.10616*, 2021.

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

[72] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Hostallero, and Yung Yi. QTRAN: learning to factorize with transformation for cooperative multi-agent reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5887–5896. PMLR, 2019.

[73] Marco Pangallo, James Sanders, Tobias Galla, and Doyne Farmer. A taxonomy of learning dynamics in 2 x 2 games, 2017.

[74] Satinder Singh, Michael Kearns, and Yishay Mansour. Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 541–548. Morgan Kaufmann Publishers Inc., 2000.

[75] Anurag Koul. A collection of multi agent environments based on OpenAI gym, 2019.

[76] Petr Šebek. Nash equilibria noncooperative games., sept 2013.

Appendix for *"A Game-Theoretic Approach to Multi-Agent Trust Region Optimization"*

Table of Contents

A MATRL Algorithm Based on PPO 1

B Independent Trust Payoff Region 2

C Proof of Theorem 1 3

D Proof of Theorem 2 4

E Environment Details 5

F Experimental Parameter Settings 6

A MATRL Algorithm Based on PPO

Algorithm 2 Multi-Agent Trust Region Learning Algorithm (PPO Based, Two-Agent Example).

Input: The initial policy parameters θ_1, θ_2, initial value function parameters ϕ_1, ϕ_2 and ϵ.

for $k \in \{0, 1, 2, \cdots\}$ **do**
  Using $\pi_1(\theta_1)$, $\pi_2(\theta_2)$ to collect trajectories $\boldsymbol{\tau}_1, \boldsymbol{\tau}_2$.
  Compute GAE reward $\hat{R}_i$ for each i.
  Compute estimated advantages $\hat{A}_1, \hat{A}_2$ based on the current value functions V_{ϕ_1}, V_{ϕ_2}.
  for $i \in \{1, 2\}$ **do**
    Compute a trust payoff region policy $\hat{\pi}_i$ using Eq. 2.
    Update the policy by maximizing the PPO-Clip objective:
    $\hat{\theta}_i = \arg\max_{\theta_i} \frac{1}{|\boldsymbol{\tau}_i|T} \sum_{\tau \in \boldsymbol{\tau}_i} \sum_{t=0}^{T} \min\left(\frac{\pi_i(a_t|s_t;\theta)}{\pi_i(a_{1,t}|s_t;\theta_i)} A_i^{\pi_1,\pi_2}(s_t, a_{1,t}, a_{2,t}), \quad g\left(\epsilon, A_i^{\pi_1,\pi_2}(s_t, a_{1,t}, a_{2,t})\right)\right),$
    where g is a clipping function.
    Fit value function by regression on mean-squared error:

$$\phi_i' = \arg\min_{\phi_i} \frac{1}{|\boldsymbol{\tau}_i| T} \sum_{\tau \in \boldsymbol{\tau}_i} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_{i,t}\right)^2$$

  end for
  Construct the meta-game $\mathcal{M}(\pi_1(\theta_1), \hat{\pi}_1(\hat{\theta}_1), \pi_2(\theta_2), \hat{\pi}_2(\hat{\theta}_2))$.
  Solve $\mathcal{M}$ and obtain meta Nash ρ_1, ρ_2.
  Compute aggregated weak stable fixed point $(\bar{\pi}_1, \bar{\pi}_2)$.
  for $i \in \{1, 2\}$ **do**
    Compute $\pi_i^{(\prime)}$ which best responses to $\bar{\pi}_{-i}$ using Eq. 6.
    Estimate the best response by importance sampling:

$$\theta_i' = \frac{\hat{\theta}_i}{|\boldsymbol{\tau}_i| T} \sum_{\tau \in \boldsymbol{\tau}_i} \sum_{t=0}^{T} g\left(\epsilon, \pi_i / \bar{\pi}_{-i}\right)$$

  end for
  $\theta_1 \leftarrow \theta_1', \theta_2 \leftarrow \theta_2'$.
end for

Output: $\pi_1(\theta_1)$, $\pi_2(\theta_2)$.

B Independent Trust Payoff Region

We use the total variation divergence, which is defined by $D_{\mathrm{TV}}(p\|q) = \frac{1}{2}\sum_j |p_j - q_j|$ for discrete probability distributions p, q [35]. $D_{\mathrm{TV}}^{\max}(\pi, \tilde{\pi})$ is defined as:

$$D_{\mathrm{TV}}^{\max}(\pi, \tilde{\pi}) = \max_s D_{\mathrm{TV}}(\pi(\cdot|s)\|\tilde{\pi}(\cdot|s)). \tag{7}$$

Based on this, we can define α-coupled policy as:

Definition B.1 (α-Coupled Policy [35]). *(π, π') is an α-coupled policy pair if it defines a joint distribution $(a, a')|s$, such that $P(a \neq a'|s) \leq \alpha$ for all s. π and π' will denote the marginal distributions of a and a', respectively.*

When the joint policy pair π_i, π_{-i} changes to π'_i, π'_{-i} and coupled with α_i and α_{-i} correspondingly:

$$\eta_i(\pi'_i, \pi'_{-i}) - \eta_i(\pi_i, \pi_{-i}) \geq A_i^{\pi_i,\pi_{-i}}(\pi'_i, \pi'_{-i}) - \frac{4\gamma\epsilon}{(1-\gamma)^2}(\alpha_i + \alpha_{-i} - \alpha_i\alpha_{-i})^2, \tag{8}$$

where

$$\epsilon = \max_{s,a_i,a_{-i}} \left|A_i^{\pi_i,\pi_{-i}}(s, a_i, a_{-i})\right|.$$

The proofs are as following:

Lemma B.1. *Given that (π_i, π'_i) and (π_{-i}, π'_{-i}) are both α-coupled policies bounded by α_i and α_{-i} respectively, for all s,*

$$\left|A_i^{\pi_i,\pi_{-i}}(s)\right| \leq 2(\alpha_i + \alpha_{-i} - \alpha_i\alpha_{-i}) \max_{s,a_{-i},a_{-i}} \left|A_i^{\pi_i,\pi_{-i}}(s, a_i, a_{-i})\right| \tag{9}$$

Proof.

$$\begin{aligned}
A_i^{\pi_i,\pi_{-i}}(s) &= \mathbb{E}_{a'_i,a'_{-i}\sim\pi'_i,\pi'_{-i}}\left[A_i^{\pi_i,\pi_{-i}}(s, a'_i, a'_{-i})\right] && (10)\\
&= \mathbb{E}_{(a_i,a'_i)\sim(\pi_i,\pi'_i),(a_{-i},a'_{-i})\sim(\pi_{-i},\pi'_{-i})}\left[A_i^{\pi_i,\pi_{-i}}(s, a'_i, a'_{-i}) - A_i^{\pi_i,\pi_{-i}}(s, a_i, a_{-i})\right] && (11)\\
&= P(a_i \neq a'_i \vee a_{-i} \neq a'_{-i}|s)\mathbb{E}_{(a_i,a'_i)\sim(\pi_i,\pi'_i),(a_{-i},a'_{-i})\sim(\pi_{-i},\pi'_{-i})}[A_i^{\pi_i,\pi_{-i}}(s, a'_i, a'_{-i}) && (12)\\
&\qquad - A_i^{\pi_i,\pi_{-i}}(s, a_i, a_{-i})] && (13)\\
&\leq (\alpha_i + \alpha_{-i} - \alpha_i\alpha_{-i}) \cdot 2 \max_{s,a_{-i},a_{-i}} \left|A_i^{\pi_i,\pi_{-i}}(s, a_i, a_{-i})\right|, && (14)
\end{aligned}$$

where $P(a_i \neq a'_i \vee a_{-i} \neq a'_{-i}|s) = 1 - (1-\alpha_i)(1-\alpha_{-i}) = \alpha_i + \alpha_{-i} - \alpha_i\alpha_{-i}$.

□

Lemma B.2. *Let (π_i, π'_i) and (π_{-i}, π'_{-i}) are α-coupled policy pairs. Then,*

$$\begin{aligned}
&\left|\mathbb{E}_{s_t\sim\pi'_i,\pi'_{-i}}\left[A_i^{\pi_i,\pi_{-i}}(s)\right] - \mathbb{E}_{s_t\sim\pi_i,\pi_{-i}}\left[A_i^{\pi_i,\pi_{-i}}(s)\right]\right|\\
&\qquad \leq 4(\alpha_i + \alpha_{-i} - \alpha_i\alpha_{-i})(1 - (1-\alpha_i)^t(1-\alpha_{-i})^t) \max_{s,a_{-i},a_{-i}} \left|A_i^{\pi_i,\pi_{-i}}(s, a_i, a_{-i})\right|
\end{aligned} \tag{15}$$

Proof. The preceding Lemma bounds the difference in expected advantage at each time step t. When $t' = 0$ indicates that π_i, π_{-i} and π'_i, π'_{-i} both agreed on all time steps less than t. By the definition of α_i, α_{-i}, $P(\pi_i, \pi_{-i} := \pi'_i, \pi'_{-i}|t = i) \geq (1-\alpha_i)(1-\alpha_{-i})$, so $P(t' = 0) \geq (1-\alpha_i)^t(1-\alpha_{-i})^t$ and $P(t' > 0) \leq 1 - (1-\alpha_i)^t(1-\alpha_{-i})^t$. We can sum over time to bind the difference between

$\eta_i(\pi'_i, \pi'_{-i})$ and $\eta_i(\pi_i, \pi_{-i})$.

$$\Big|\eta_i(\pi'_i, \pi'_{-i}) - L_i^{\pi_i,\pi_{-i}}(\pi'_i, \pi'_{-i})\Big| = \sum_{t=0}^{\infty} \gamma^t \Big|\mathbb{E}_{s_t \sim \pi'_i, \pi'_{-i}}\big[A_i^{\pi_i,\pi_{-i}}(s)\big] - \mathbb{E}_{s_t \sim \pi_i, \pi_{-i}}\big[A_i^{\pi_i,\pi_{-i}}(s)\big]\Big| \tag{16}$$

$$\leq \sum_{t=0}^{\infty} \gamma^t \cdot 4\epsilon(\alpha_i + \alpha_{-i} - \alpha_i\alpha_{-i})(1 - (1-\alpha_i)^t(1-\alpha_{-i})^t) \tag{17}$$

$$= 4\epsilon(\alpha_i + \alpha_{-i} - \alpha_i\alpha_{-i})\Big(\frac{1}{1-\gamma} - \frac{1}{1-\gamma(1-\alpha_i)(1-\alpha_{-i})}\Big) \tag{18}$$

$$= \frac{4\epsilon(\alpha_i + \alpha_{-i} - \alpha_i\alpha_{-i})^2}{(1-\gamma)(1-\gamma(1-\alpha_i)(1-\alpha_{-i}))} \tag{19}$$

$$\leq \frac{4\epsilon(\alpha_i + \alpha_{-i} - \alpha_i\alpha_{-i})^2}{(1-\gamma)^2}, \tag{20}$$

where $\epsilon = \max_{s,a_i,a_{-i}} \big|A_i^{\pi_i,\pi_{-i}}(s, a_i, a_{-i})\big|$. □

Note that

$$L_i^{\pi_i,\pi_{-i}}(\pi'_i, \pi'_{-i}) = \eta_i(\pi_i, \pi_{-i}) + \sum_s \rho^{\pi_i,\pi_{-i}}(s) \sum_{a_i} \pi'_i(a_i|s) \sum_{a_{-i}} \pi'_{-i}(a_{-i}|s) A_i^{\pi_i,\pi_{-i}}(s, a_i, a_{-i}). \tag{21}$$

Then, we can have

$$\eta_i(\pi'_i, \pi'_{-i}) - \eta_i(\pi_i, \pi_{-i}) \geq A_i^{\pi_i,\pi_{-i}}(\pi'_i, \pi'_{-i}) - \frac{4\gamma\epsilon}{(1-\gamma)^2}(\alpha_i + \alpha_{-i} - \alpha_i\alpha_{-i})^2. \tag{22}$$

C Proof of Theorem 1

At each iteration, denote $\nabla_i g_i = \nabla_{\pi_i} g_i^{\pi_i,\pi_{-i}}$ and $\nabla_{i,-i} g_i = \nabla_{\pi_i}\nabla_{\pi_{-i}} g_i^{\pi_i,\pi_{-i}}$ for each i. Consider the simultaneous gradient $\boldsymbol{\xi}$ of the expected advantage gains and the corresponding Hessian H:

$$\boldsymbol{\xi}(\pi_i, \pi_{-i}) = (\nabla_i g_i, \nabla_{-i} g_{-i}), \tag{23}$$

$$H = \nabla\xi = \begin{pmatrix} \nabla_{i,i} g_i & \nabla_{i,-i} g_i \\ \nabla_{-i,i} g_{-i} & \nabla_{-i,-i} g_{-i} \end{pmatrix}. \tag{24}$$

For a restricted underlying game, where policy space is bounded: $\pi_i \in [\pi_i, \hat{\pi}_i]$. Assume π_i is the linear mixture of $\pi_i, \hat{\pi}_i$, and $\bar{\pi}_i = \rho_i\pi_i + (1-\rho_i)\hat{\pi}_i$, where $\rho_i \in [0,1]$. Therefore, we can re-write the $g_i^{\pi_i,\pi_{-i}}(\pi_i, \pi_{-i})$ in the form of:

$$g_i^{\pi_i,\pi_{-i}}(\pi_i, \pi_{-i}) = g_i^{\pi_i,\pi_{-i}}(\rho_i, \rho_{-i}) = \rho_i(1-\rho_{-i})g_i^{i,-\hat{i}} + (1-\rho_i)\rho_{-i}g_i^{\hat{i},-i} + (1-\rho_i)(1-\rho_{-i})g_i^{\hat{i},-\hat{i}}. \tag{25}$$

Then we have:

$$\nabla_i g_i(\rho_{-i}) = (1-\rho_{-i})g_i^{i,-\hat{i}} - \rho_{-i}g_i^{\hat{i},-i} - (1-\rho_{-i})g_i^{\hat{i},\hat{-i}}, \tag{26}$$

and $\boldsymbol{\xi}(\pi_i, \pi_{-i}) = \boldsymbol{\xi}(\rho_i, \rho_{-i})$. Given a meta Nash policy pair $(\bar{\pi}_i, \bar{\pi}_{-i})$, where $\bar{\pi}_i = \bar{\rho}^i\pi_i + (1-\bar{\rho}^i)\hat{\pi}_i$, according to the Nash definition, we have:

$$\begin{pmatrix} \bar{\rho}_i \\ 1-\bar{\rho}_i \end{pmatrix}^T \begin{pmatrix} g_i^{i,-i} & g_i^{i,-\hat{i}} \\ g_i^{\hat{i},-i} & g_i^{\hat{i},-\hat{i}} \end{pmatrix} \begin{pmatrix} \bar{\rho}_{-i} \\ 1-\bar{\rho}_{-i} \end{pmatrix} \geq \begin{pmatrix} \rho_i \\ 1-\rho_i \end{pmatrix}^T \begin{pmatrix} g_i^{i,-i} & g_i^{i,-\hat{i}} \\ g_i^{\hat{i},-i} & g_i^{\hat{i},-\hat{i}} \end{pmatrix} \begin{pmatrix} \bar{\rho}_{-i} \\ 1-\bar{\rho}_{-i} \end{pmatrix}, \tag{27}$$

which implies:

$$\begin{aligned} (\bar{\rho}_i - \rho_i)\nabla_i g_i(\bar{\rho}_{-i}) \geq 0, \quad & \bar{\rho}_i, \forall \rho_{-i} \in [0,1], \\ (\bar{\rho}_{-i} - \rho_{-i})\nabla_{-i} g_{-i}(\bar{\rho}_i) \geq 0, \quad & \bar{\rho}_i, \forall \rho_{-i} \in [0,1]. \end{aligned} \tag{28}$$

When $\bar{\rho}_i, \bar{\rho}_{-i} \in (0, 1)$ in accordance with the Nash condition in Eq. 28, $\nabla_i g_i(\bar{\rho}_{-i}) = \nabla_{-i} g_{-i}(\bar{\rho}_i) = 0$. It shows that $(\bar{\pi}_i, \bar{\pi}_{-i})$ is a fixed point due to $\boldsymbol{\xi}(\bar{\pi}_i, \bar{\pi}_{-i}) = \boldsymbol{\xi}(\bar{\rho}_i, \bar{\rho}_{-i}) = \mathbf{0}$. For the boundary case, where $\bar{\rho}_i$ or $\bar{\rho}_{-i} \in \{0, 1\}$, because they are constrained to the unit square $[0, 1] \times [0, 1]$, the gradients on the boundaries of the unit square are projected onto the unit square, which means additional points of zero gradient exist. In other words, $\nabla_i g_i$ and $\nabla_{-i} g_{-i}$ are still equal to zero in boundary case, and the $(\bar{\pi}_i, \bar{\pi}_{-i})$ is a fixed point in both cases.

Next, we determine what types of the fixed point that $(\bar{\pi}_i, \bar{\pi}_{-i})$ belongs to. According to the Eq. 24, we have the exact Hessian Matrix for the restricted game:

$$H = \nabla \xi = \begin{pmatrix} 0 & g_i^{\hat{i},-\hat{i}} - g_i^{i,-\hat{i}} - g_i^{\hat{i},-i} \\ g_{-i}^{\hat{i},-\hat{i}} - g_{-i}^{i,-\hat{i}} - g_{-i}^{\hat{i},-i} & 0 \end{pmatrix} \tag{29}$$

The eigenvalue λ of H can be computed:

$$\lambda^2 - \mathrm{Tr}(H)\lambda + \det(H) = \lambda^2 - (g_i^{\hat{i},-\hat{i}} - g_i^{i,-\hat{i}} - g_i^{\hat{i},-i})(g_{-i}^{\hat{i},-\hat{i}} - g_{-i}^{i,-\hat{i}} - g_{-i}^{\hat{i},-i}) = 0 \tag{30}$$

Denotes $\bar{g}_i := g_i^{\hat{i},-\hat{i}} - g_i^{i,-\hat{i}} - g_i^{\hat{i},-i}$, we have $\boldsymbol{\lambda} = \pm\sqrt{\bar{g}_i \bar{g}_{-i}}$. Therefore, we can have following cases for the fixed point $(\bar{\rho}_i, \bar{\rho}_{-i})$:

1. Fully cooperative games: $\bar{g}_i \leq 0, \bar{g}_{-i} \leq 0$, then $H(\bar{\rho}_i, \bar{\rho}_{-i}) \preceq 0$, which means $(\bar{\rho}_i, \bar{\rho}_{-i})$ is a stable fixed point as we are maximizing the objective.
2. Fully competitive games: $\bar{g}_i > 0, \bar{g}_{-i} < 0$ or $\bar{g}_i < 0, \bar{g}_{-i} > 0$, all $\boldsymbol{\lambda}$ have two pure imaginary eigenvalues with zero real part, where $(\bar{\rho}_i, \bar{\rho}_{-i})$ is a saddle point.
3. General-sum games: they are in-between the cooperative and competitive games, which means $(\bar{\rho}_i, \bar{\rho}_{-i})$ can be either stable fixed point or saddle point.

Because we assume $\hat{\pi}_i$ monotonically improved compared to π_i, then even in zero-sum case, there is at least one negative value in $\bar{g}_i$ and $\bar{g}_{-i}$. Therefore, in all the situations, $(\bar{\rho}_i, \bar{\rho}_{-i})$ is not unstable, and could be a stable point or saddle point. We define them as a weak stable fixed point. It also has a tighter lower bound than the independent trust region improvement seen in Remark C.1:

Remark C.1. *Let (ρ_i, ρ_{-i}) be a Nash equilibrium of the policy-space meta-game $\mathcal{M}(\pi_i, \hat{\pi}_i, \pi_{-i}, \hat{\pi}_{-i})$, which is used for computing the linear mixture policies $\bar{\pi}_i, \bar{\pi}_{-i}$. For simplicity, define $\bar{\rho}_i = 1 - \rho_i$, then we have the payoff improvement lower bound for $\bar{\pi}_i, \bar{\pi}_{-i}$:*

$$\eta_i(\bar{\pi}_i, \bar{\pi}_{-i}) - \eta_i(\pi_i, \pi_{-i}) \geq g_i^{\pi_i, \pi_{-i}}(\bar{\pi}_i, \bar{\pi}_{-i}) - \frac{4\gamma\epsilon_i}{(1-\gamma)^2}(\alpha_i \bar{\rho}_i + \alpha_{-i}\bar{\rho}_{-i} - \alpha_i \alpha_{-i} \bar{\rho}_i \bar{\rho}_{-i})^2, \tag{31}$$

that is a tighter lower bound compared with Theorem 1.

Finally, we obtain MATRL as follows: First, an agent i collects a set of trajectories using its current policy π_i by independent play with other agents. Then a predicted policy $\hat{\pi}_i$ can be estimated using the single-agent trust region methods, which has a trust payoff improvement against the other agents' current policy π_{-i}. However, this trust payoff improvements would not benefit convergence requirements for the multi-agent system due to other agents adaptive learning. To solve this problem, we approximate a n-agent two-action meta-game in policy-space by reusing the trajectories from the last TPR step. In this game, each agent i has two pure strategies: choosing the *current policy* π_i or *predicted policy* $\hat{\pi}_i$ and the corresponding payoffs are the expected advantages (defined in Eq. 3) of the joint policy pairs. By constructing such a meta-game, we transform a complex multi-agent interactions problem into game-theoretic analysis concerning the underlying game restricted in $[\pi_i, \hat{\pi}_i]$. Then we can obtain a weak stable fixed point as TSR within the TPR by solving the meta-game,. When the fixed point is a saddle point we then take the best response to the weak stable fixed point to get the next iteration's policies. This encourages exploration and avoid stagnation at an unexpected saddle point.

D Proof of Theorem 2

Let the objectives $\eta_i(\pi_1, \cdots, \pi_n)$ of agents are twice continuously differentiable, in which the agents with parameters $\theta = \left(\theta^1, \ldots, \theta^n\right)$. Denote ξ as the simultaneous gradient of game, we can obtain the

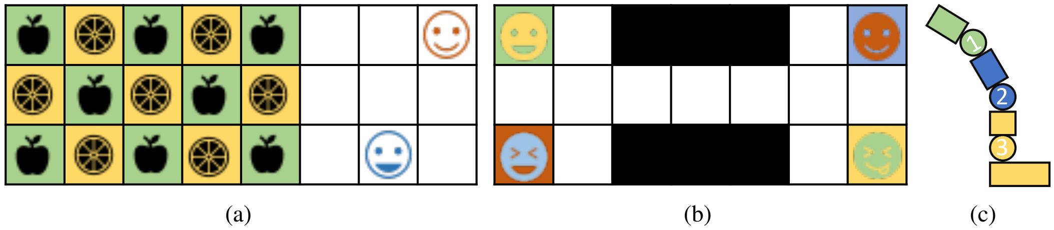


Figure 6: Multi-agent discrete and continuous action tasks: (a) 2-agent checker (discrete), (b) 4-agent switch (discrete), (c) 3-agent MuJoCo hopper (continious).

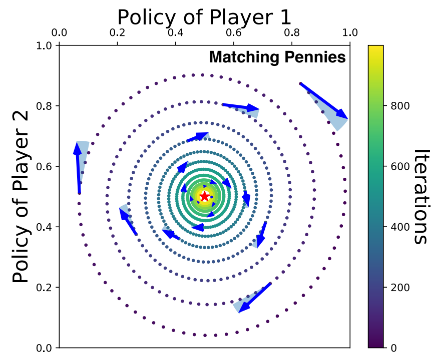


Figure 7: Learning the dynamics of MATRL in a matching pennies (MP) game. The blue arrow is the gradient direction, and the pale blue area is the STR.

corresponding Hession of the game:

$$H = \nabla \xi = \begin{pmatrix} \nabla_{11}\eta_1 & \cdots & \nabla_{1n}\eta_1 \\ \vdots & \ddots & \vdots \\ \nabla_{n1}\eta_n & \cdots & \nabla_{nn}\eta_n \end{pmatrix}.$$

Let H_o is the matrix of anti-diagonal blocks of H (Hessian of the game), and α is step-size. For MATRL, we have the updating gradient $(I - \rho\alpha H_o)\,\xi$, where ρ is a ratio determined by meta-game Nash to dynamically adjust the step-size at each iteration. Then the iterative procedure:

$$F(\theta) = \theta + \alpha(I - \rho\alpha H_o)\xi(\theta).$$

Assume $\bar{\theta}$ is a fixed point if $\xi(\bar{\theta}) = 0$, and denote $X := (I - \rho\alpha H_o)$, then we have:

$$\nabla[X\xi](\bar{\theta}) = \nabla X(\bar{\theta})\xi(\bar{\theta}) + X(\bar{\theta})\nabla\xi(\bar{\theta}) = XH(\bar{\theta})$$

is negative stable (if all its eigenvalues of X have negative real part) according to Theorem 1, namely has eigenvalues $a_k + ib_k$ with $a_k < 0$. It means

$$\nabla F(\bar{x}) = I + \alpha\nabla[X\xi](\bar{x})$$

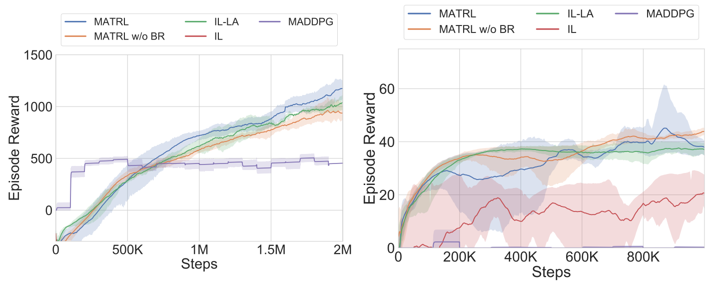
has eigenvalues $1 + \alpha a_k + i\alpha b_k$ in the a small circle with radius $\epsilon \geq 0$:

$$|1 + \alpha a_k + i\alpha b_k|^2 < 1 \iff \quad 0 < \alpha < \frac{-2a_k}{a_k^2 + b_k^2},$$

which is always possible for $a_k < 0$. Then it is sufficient to prove the converges locally to $\bar{\theta}$ with ϵ error for α sufficiently small according to Ostrowski's Theorem [45].

E Environment Details

Random 2 $\times$ 2 Matrix Games. We created a generator of 2×2 matrix games based on the category provided by [73]. Coordination games have characteristics enabling one agent to improve the payoff without decreasing the payoff of the other agent. Anti-coordination games are ones where one agent improves the payoff while the other agent's payoff decreases. Both coordination and anti-coordination games can have two pure NEs and one mixed strategy NE. In cyclic games, the action selections of



(a) 2-Agent Ant.

(b) 2-Agent Swimmer.

Figure 8: Learning curves of 2-Agent Ant and 2-Agent Swimmer MuJoCo tasks.

agents that is based on their actions will form a cycle, ensuring that there is no pure NE in the game. Instead only mixed strategy NE will be found.

Grid World Games. In two-player checker, as shown in Fig. 6a, there is one sensitive player who gets reward 5 when they collect an apple and 5 when they collect a lemon; a less sensitive player gets 1 for apple and 1 for lemon. The learning goal is to let the sensitive player get apples and the other one get lemons to have a higher total reward. In four-player switch, as shown in Fig. 6b, to reach the targets, agents need to figure out a way to go through a narrow corridor. The agent gets -1 for taking each step and 5 when arriving at a target. Four-player switch uses the same map as two-player switch, where two agents start from the left side and the others from the right side to go through the corridor to reach the targets. With more agents in four-player switch, learning becomes more challenging. MATRL agents achieved higher total rewards compared to baseline algorithms within the same number of steps.

Multi-Agent MuJoCo Tasks. We used the three-agent Hopper environment described in [31], and Fig. 6c, where three agents control three joints of the robot and learn to cooperate to move forward as far as possible. The agent is rewarded by the number of time steps that they move without falling. Each agent has 3 continuous output values as the action, and all the agents have a full observation of the states of size 17. We use the same hyper-parameters for MATRL, MATRL w/o BR, and IL-PP. For MADDPG agent, we use the hyper-parameters described in the paper [31].

Multi-Agent Atari Game. The pong game is a multi-agent Atari version[4] of table tennis Two players must prevent a ball from whizzing past their paddles and allowing their opponent to score. The game ends when one side earns 21 points.

F Experimental Parameter Settings

For all the tasks, the most important hyper-parameters are learning rate/step size, the number of update steps, batch size and value, policy loss coefficient. Appropriate learning rate and update steps plus larger batch size give a more stable learning curve. And for different environments, policy and value network loss coefficients that keep two losses at the same scale are essential in improving the learning result and speed. Also, for meta-game construction and best response update where we use the importance ratio to do estimation, a clipping factor of the ration is vital to achieving a stable and monotonic improving result. The followings are the detailed parameter settings for each task.

Matrix Game and Random 2×2 **Matrix Games**. The hyper-parameters settings for MATRL, IGA-PP, and WoLF are listed in Table 2. As shown in Fig. 8, we also listed the additional convergence analysis in classical Chicken and Prisoners' Dilemma Games, which demonstrate good convergence performance of MATRL on both games. For MATRL, we have the KL-divergence coefficient as an extra hyper-parameter to add the KL-divergence as part of the loss in policy updating. And for the baseline algorithm WoLF, we give the real NE of the game as part of the parameters. In all the games, all the algorithms shared the same initial policy values $[0.9, 0.1]$ for player 1 and $[0.2, 0.8]$ for player 2.

[4] https://github.com/PettingZoo-Team/Multi-Agent-ALE

Table 2: Hyper-parameter settings in 2×2 matrix games.

Settings	Value	Description
Common Settings		
Initial policies 1	$[0.9, 0.1]$	The initial policy values for player 1
Initial policies 2	$[0.2, 0.8]$	The initial policy values for player 2
MATRL Settings		
Best Response learning rate	0.03	The learning rate for the best response step
KL coefficient	100	The KL-Divergence coefficient in policy loss
WoLF Settings		
Learning Rate Maximum	0.06	The maximum learning rate for WoLF learn fast agent
Learning Rate Minimum	0.02	The minimum learning rate for WoLF win agent

Table 3: MATRL hyper-parameter settings in grid worlds.

Common Settings	Value	Description
Policy Learning Rate	0.002	Optimizer learning rate.
Batch Size	2000	Number of data point for each update.
Gamma	0.99	Long Term Discount factor.
Hidden Dimension	128	Size of hidden states.
Number of Hidden Layers	2	Number of Hidden layers.
Nash Equilibrium Solver Method	CMAES	The method for finding the Nash equilibrium of meta-game
Neural Network	MLP	The neural network architecture for policy and critic
Policy Update Iterations	10	Number of gradient steps for each batch of update.
Best Response Learning Rate	0.002	The learning rate for best response step
Best Response Interactions	5	Number of gradient steps for best response step
KL Coefficient	0.001	The KL divergence coefficient in calculating loss
Entropy Coefficient	0.05	The entropy coefficient in calculating loss
Policy Ratio Clip	0.1	The clip value for policy ratio
Best Response Importance Ratio Clip	0.1	The clip value for best response importance weight
2 Player Switch		
Value loss Coefficient	0.01	The value loss is larger than policy loss
2 Player Checker		
Value loss Coefficient	1.0	The value loss is at same scale as policy loss
4 Player Switch		
Value loss Coefficient	0.01	The value loss is larger than policy loss

Grid World Games and Multi-agent Continuous Control Task. The hyper-parameters settings for MATRL are given in Table 3. We used the same hyper-parameters for MATRL, MATRL w/o BR, IL-PP, and IL. The only difference is whether to use Best Response and the meta-game or not. We used Leaky ReLU as the activation function for both policies and value networks. For the training, we used paralleled workers to collect experience data and update the network weights separated then synchronize all the works to have the final updated weights. We used different value loss and policy loss coefficients to balance the weights of two losses. For the Switch games, we used small value loss coefficients because the value loss is between $[0 - 10]$ while the absolute value policy loss is smaller than $1e - 2$. For the Checker game, the value loss and policy loss are at the same scale between $[1e - 4, 1e - 2]$. Also, we added entropy loss and KL loss to encourage exploration and limit the policy update for each step. We used [76] as the Nash equilibrium solver for finding the meta-game Nash. The Nash solver is CMAES for all the experiments. If not particularly indicated, all the baselines use common settings as listed in Table 3. VDN, QMIX use common individual action-value networks as those used by MATRL; each consists of two 128-width hidden layers. We includes more experiment result on 4-agent ant task multi-agent MuJoCo task in Fig. 8a, which also demonstrate the superior performance of MATRL compared to other settings. The specialized parameter settings for each algorithm are provided in Table 4 and 5:

Multi-agent Atari Pong. The hyper-parameters setting for MATRL are listed in Table 6. We used the same hyper-parameters for MATRL and IL. We take the raw pixel input from the Atari environment, and we processed it with a convolution network, which has filter sizes [8,4,3], kernel sizes (3,3,3), and stride sizes [4,2,1] and "VALID" as padding. Then we pass the processed embedding to a 2 layer fully connected network to get the policy.

Table 4: Hyper-parameter settings for baseline algorithms in grid worlds.

SETTINGS	VALUE	DESCRIPTION
VDN		
MONOTONE NETWORK LAYER	2	LAYER NUMBER OF MONOTONE NETWORK.
MONOTONE NETWORK SIZE	128	HIDDEN LAYER SIZE OF MONOTONE NETWORK.
TARGET NETWORK UPDATE INTERVAL	200	NUMBER OF ITERATIONS BETWEEN EACH TARGET NETWORK UPDATE
LEARNER	DOUBLE-Q LEARNER	THE ALGORITHMS FOR EACH AGENT
QMIX		
JOINT ACTION-VALUE NETWORK LAYER	2	LAYER NUMBER OF JOINT ACTION-VALUE NETWORK.
JOINT ACTION-VALUE NETWORK SIZE	128	HIDDEN LAYER SIZE OF JOINT ACTION-VALUE NETWORK.
LEARNER	DOUBLE-Q LEARNER	THE ALGORITHMS FOR EACH AGENT

Table 5: Hyper-parameter settings in multi-agent MuJoCo hopper.

SETTINGS	VALUE	DESCRIPTION
MATRL AND ITS VARIANTS		
AGENT ALGORITHM	PPO	THE LEARNING ALGORITHM FOR AGENT
NETWORK	2 LAYER MLP [128, 128]	THE NETWORK ARCHITECTURE AND SIZE FOR THE PPO AGENT
LEARNING RATE	0.002	LEARNING RATE FOR AGENTS
BATCH IZE	4000	BATCH SIZE FOR ONE UPDATE
VALUE LOSS COEFFICIENT	0.001	THE VALUE LOSS COEFFICIENT IN TOTAL LOSS
POLICY LOSS COEFFICIENT	100	THE POLICY LOSS COEFFICIENT IN TOTAL LOSS
POLICY UPDATE ITERATIONS	10	NUMBER OF GRADIENT STEPS FOR EACH BATCH OF UPDATE.
BEST RESPONSE LEARNING RATE	0.002	THE LEARNING RATE FOR BEST RESPONSE STEP
BEST RESPONSE INTERACTIONS	5	NUMBER OF GRADIENT STEPS FOR BEST RESPONSE STEP
ENTROPY COEFFICIENT	0.05	THE ENTROPY COEFFICIENT IN TOTAL LOSS
KL-DIVERGENCE COEFFICIENT	0.01	THE KL-DIVERGENCE COEFFICIENT IN TOTAL LOSS
GAMMA	0.99	DISCOUNT FACTOR
MADDPG		
NETWORK	2 LAYER MLP [300, 300]	THE NETWORK ARCHITECTURE AND SIZE FOR THE PPO AGENT
LEARNING RATE	0.001	LEARNING RATE FOR AGENTS
BATCH SIZE	100	BATCH SIZE FOR ONE UPDATE
UPDATE INTERVAL	100	UPDATE THE NETWORK EVERY 100 TIME STEPS
PRE-TRAIN TIMETEPS	10000	NUMBER OF TIME STEPS BEFORE NETWORK UPDATE
GAMMA	0.99	DISCOUNT FACTOR
COMIX		
HYPER-NETWORK LAYER	2	LAYER NUMBER OF HYPER-NETWORK.
HYPER-NETWORK SIZE	64	HIDDEN LAYER SIZE OF HYPER-NETWORK.
ACT NOISE	200	STDDEV FOR GAUSSIAN EXPLORATION NOISE ADDED TO POLICY AT TRAINING TIME.
LEARNER	DOUBLE-Q LEARNER	THE ALGORITHMS FOR EACH AGENT

Table 6: Hyper-parameter settings in multi-agent pong Atari.

SETTINGS	VALUE	DESCRIPTION
MATRL AND ITS VARIANTS		
AGENT ALGORITHM	PPO	THE LEARNING ALGORITHM FOR AGENT
NETWORK	3 LAYER CNN, 2 LAYER FC	THE NETWORK ARCHITECTURE AND SIZE FOR THE PPO AGENT
LEARNING RATE	0.002	LEARNING RATE FOR AGENTS
BATCH IZE	4000	BATCH SIZE FOR ONE UPDATE
VALUE LOSS COEFFICIENT	0.1	THE VALUE LOSS COEFFICIENT IN TOTAL LOSS
POLICY LOSS COEFFICIENT	10	THE POLICY LOSS COEFFICIENT IN TOTAL LOSS
POLICY UPDATE ITERATIONS	10	NUMBER OF GRADIENT STEPS FOR EACH BATCH OF UPDATE.
BEST RESPONSE LEARNING RATE	0.002	THE LEARNING RATE FOR BEST RESPONSE STEP
BEST RESPONSE INTERACTIONS	5	NUMBER OF GRADIENT STEPS FOR BEST RESPONSE STEP
ENTROPY COEFFICIENT	0.05	THE ENTROPY COEFFICIENT IN TOTAL LOSS
KL-DIVERGENCE COEFFICIENT	0.01	THE KL-DIVERGENCE COEFFICIENT IN TOTAL LOSS
GAMMA	0.99	DISCOUNT FACTOR