Competitive Policy Optimization

Manish Prajapat[1,3] **Kamyar Azizzadenesheli**[2] **Alexander Liniger**[1] **Yisong Yue**[3] **Anima Anandkumar**[3]

[1]ETH Zurich,
[2]Purdue University,
[3]California Institute of Technology

Abstract

A core challenge in policy optimization in competitive Markov decision processes is the design of efficient optimization methods with desirable convergence and stability properties. We propose competitive policy optimization (CoPO), a novel policy gradient approach that exploits the game-theoretic nature of competitive games to derive policy updates. Motivated by the competitive gradient optimization method, we derive a bilinear approximation of the game objective. In contrast, off-the-shelf policy gradient methods utilize only linear approximations, and hence do not capture players' interactions. We instantiate CoPO in two ways: (i) competitive policy gradient, and (ii) trust-region competitive policy optimization. We theoretically study these methods, and empirically investigate their behavior on a set of comprehensive, yet challenging, competitive games. We observe that they provide stable optimization, convergence to sophisticated strategies, and higher scores when played against baseline policy gradient methods.

1 INTRODUCTION

Reinforcement learning (RL) in competitive Markov decision processes CoMDP [Filar and Vrieze, 2012] is the study of competitive players, sequentially making decisions in an environment. In CoMDPs, the competing agents (players) interact with each other within the environment, and through their interactions, learn how to develop their behavior and improve their policy. In this paper, we consider the rich and fundamental class of zero-sum two-player games.

A core challenge in CoMDP is to design optimization procedures with desirable convergence and stability properties. Policy gradient (PG) is a prominent RL approach that is widely used in single agent optimization and derives policy update using the first order (linear) approximation of the objective function [Robbins and Monro, 1951, Aleksandrov et al., 1968, Sutton et al., 2000]. A straightforward extension of conventional single-agent PG approaches to two-player min-max games results in the gradient descent ascent (GDA) PG algorithm. This approximation is linear in agents' parameters and does not take their interaction into account. Therefore, GDA directly optimizes the policy of each agent, assuming the policy of the opponent is fixed which some times leads to divergence even in simple scenarios and hence considered undesirable in competitive optimization.

We propose a new paradigm, competitive policy optimization (CoPO) for solving two-player CoMDPs. CoPO exploits the game-theoretic and competitive nature of CoMDPs, and, inspired by the competitive gradient descent approach [Schaefer and Anandkumar, 2019], deploys a local bilinear approximation of the game objective to derive policy updates. This local bilinear approximation can be viewed as the simultaneous two-player generalization of the local linear approximation used in single-agent policy gradient approaches (holding the other agent's policy fixed). To compute the policy updates, CoPO computes the Nash equilibrium of the local bilinear approximation of the game objective. In CoPO, each agent derives its update with the consideration of what the other agent's current move and moves in the future time steps might be. In addition, each agent considers how the environment, as the result of the agents' current and future moves, evolves in favor of each agent. Therefore, each agent hypothesizes about what the other agent's and the environment's responses would be, resulting in the *recursive reasoning* in game theory [Keynes, 2018] and temporal recursion in CoMDPs.

We instantiate CoPO in two ways to arrive at practical algorithms. We propose competitive policy gradient (CoPG), a novel PG algorithm that exploits value functions and the structure of CoMDPs to efficiently obtain policy updates. We further extend our approach to the case where each agent does not have direct access to the opponent's policy parameters, and must (approximate) it. We also propose trust region

Accepted for the 37th Conference on Uncertainty in Artificial Intelligence (UAI 2021).

competitive policy optimization (TRCoPO), a novel trust region based PG method [Schulman et al., 2015]. TRCoPO updates agents' parameters simultaneously by deriving the Nash equilibrium of a bilinear (in contrast to linear approximation in off-the-shelf trust region methods) approximation to the surrogate objective within a defined trust region in the parameter space.

We empirically validate our approach in several settings. We construct the main empirically study on competitive games, such as Markov soccer, Linear dynamical systems, and Racing cars. We show in all these environments that CoPO leads to superior policies. We observe many cases where standard policy gradient approaches do not exhibit stable learning behavior and can diverge. In our case studies, we show that CoPO can be applied to self-play setting, where an agent is playing against itself. We further show that CoPO can be applied to improve performance in other competitive algorithms such as generative adversarial imitation learning (GAIL) (where one player is the policy and the other is the discriminator) [Ho and Ermon, 2016]. We further extend our case study and show that CoPO remains effective even when one needs to learn a model of the opponent's policy rather than having direct access.

2 PRELIMINARIES

A two player CoMDP is a tuple of $\langle \mathcal{S}, \mathcal{A}^1, \mathcal{A}^2, \mathcal{R}, \mathcal{T}, \mathcal{P}, \gamma \rangle$, where $\mathcal{S}$ is the state space, $s \in \mathcal{S}$ is a state, for player $i \in \{1,2\}$, $\mathcal{A}^i$ is the player i's action space with $a^i \in \mathcal{A}^i$. $\mathcal{R}$ is the reward kernel with probability distribution $R(\cdot|s,a^1,a^2)$ and mean function $r(s,a^1,a^2)$ on $\mathbb{R}$. For a probability measure $\mathcal{P}$, p denotes the probability distribution of initial state, and for the transition kernel $\mathcal{T}$, $T(s'|s,a^1,a^2)$ is the distribution of successive state s' after taking actions a^1, a^2 simultaneously at state s, with discount factor $\gamma \in [0,1]$. We consider episodic environments with reachable absorbing state s_T almost surely in finite time. An episode starts at $s_0 \sim p$, and at each time step $k \geq 0$ at state s_k, each player i draws its action a^i_k according to policy $\pi(a^i_k|s_k;\theta^i)$ parameterized by $\theta^i \in \Theta^i$, where $\Theta^i \subset \mathbb{R}^l$ is a compact metric space. Players $1,2$ receive $(r_k, -r_k)$ with $r_k \sim R(s_k, a^1_k, a^2_k)$, and the environment evolves to a new state s_{k+1}. A realization of this stochastic process is a trajectory $\tau = \big((s_k, a^1_k, a^2_k, r_k)_{k=0}^{|\tau|-1}, s_{|\tau|}\big)$, an ordered sequence with random length $|\tau|$, where $|\tau|$ is determined by episode termination time and state $s_{|\tau|} = s_T$. Let $f(\tau;\theta^1,\theta^2)$ denote the probability distribution of the trajectory τ following players' policies $\pi(\theta^i)$,

$$f(\tau;\theta^1,\theta^2) = p(s_0)\prod_{k=0}^{|\tau|-1}\pi(a^1_k|s_k;\theta^1)\pi(a^2_k|s_k;\theta^2) R(r_k|s_k,a^1_k,a^2_k)T(s_{k+1}|s_k,a^1_k,a^2_k). \quad (1)$$

For $\mathsf{R}(\tau) = \sum_{k=0}^{|\tau|}\gamma^k r(s_k,a^1_k,a^2_k)$, the Q-function, V-functions, and game objective are defined,

$$Q(s_k,a^1_k,a^2_k;\theta^1,\theta^2)=\mathbb{E}_{\tau\sim f(\cdot;\theta^1,\theta^2)}\Big[\sum_{j=k}^{|\tau|-1}\gamma^{j-k}r(s_j,a^1_j,a^2_j)|s_k,a^1_k,a^2_k\Big],$$

$$V(s_k;\theta^1,\theta^2)=\mathbb{E}_{\tau\sim f(\cdot;\theta^1,\theta^2)}\Big[\sum_{j=k}^{|\tau|-1}\gamma^{j-k}r(s_j,a^1_j,a^2_j)|s_k\Big],$$

$$\eta(\theta^1,\theta^2) = \int_\tau f(\tau;\theta^1,\theta^2)\mathsf{R}(\tau)d\tau \quad (2)$$

We assume V, Q, and η are differentiable and bounded in (Θ^1,Θ^2) and for f on (Θ^1,Θ^2), $D_{\theta^i}f = \frac{\partial}{\partial\theta'^i}f(\theta'^1,\theta'^2)\big|_{(\theta'^1,\theta'^2)=(\theta^1,\theta^2)}$, and $D_{\theta^i\theta^j}f = \frac{\partial}{\partial\theta'^i}\big(\frac{\partial}{\partial\theta'^j}f(\theta'^1,\theta'^2)\big)\big|_{(\theta'^1,\theta'^2)=(\theta^1,\theta^2)}$, for $i,j\in\{1,2\}$.

3 COMPETITIVE POLICY OPTIMIZATION

Player 1 aims to maximize the game objective η Eq. (2), and player 2 aims to minimize it, i.e., simultaneously solving for $\max_{\theta^1}\eta(\theta^1,\theta^2)$ and $\min_{\theta^2}\eta(\theta^1,\theta^2)$ respectively with,

$$\theta^{1*} \in \underset{\theta^1\in\Theta^1}{\operatorname{argmax}}\,\eta(\theta^1,\theta^2), \text{ and } \theta^{2*} \in \underset{\theta^2\in\Theta^2}{\operatorname{argmin}}\,\eta(\theta^1,\theta^2). \quad (3)$$

As discussed in the introduction, a straightforward generalization of PG methods to CoMDP, results in GDA (Alg.1). Given players' parameters (θ^1,θ^2), GDA prescribes to optimize a linear approximation of the game objective in the presence of a regularization for the policy updates,

$$\theta^1 \leftarrow \theta^1 + \underset{\Delta\theta^1:\Delta\theta^1+\theta^1\in\Theta^1}{\operatorname{argmax}}\Delta\theta^{1\top}D_{\theta^1}\eta - \frac{1}{2\alpha}||\Delta\theta^1||^2, \text{ and}$$

$$\theta^2 \leftarrow \theta^2 + \underset{\Delta\theta^2:\Delta\theta^2+\theta^2\in\Theta^2}{\operatorname{argmin}}\Delta\theta^{2\top}D_{\theta^2}\eta + \frac{1}{2\alpha}||\Delta\theta^2||^2, \quad (4)$$

where α represent the step size. The parameter updates in Eq. 4 result in greedy updates along the directions of maximum change, assuming the other player stays constant. These updates are myopic, and ignore the agents' interactions. In other words, player 1 does not take player's 2 potential move into consideration and vice versa. While GDA might be an approach of interest in decentralized CoMDP, it mainly falls short in the problem of competitive and centralized optimization in a priori unknown CoMDPs, i.e., the focus of this work. In fact, this behaviour is far from optimal and is shown to diverge in many simple cases e.g. plain bilinear or linear quadratic games [Schaefer and Anandkumar, 2019, Mazumdar et al., 2019]. While single agent policy gradient methods generalize gradient descent [Robbins and Monro, 1951] to single player RL settings, in this paper, we generalize competitive gradient descent [Schaefer and Anandkumar, 2019] to zero-sum RL settings.

We propose competitive policy optimization CoPO, a policy gradient approach for optimization in unknown CoMDPs.

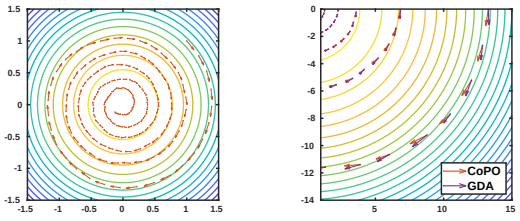
In contrast to standard PG methods, such as GDA, CoPO considers a bilinear approximation of the game objective, and takes the interaction between players into account. Following the competitive gradient updates, CoPO incorporates the game theoretic nature of the CoMDP optimization and derives parameter updates through finding the Nash equilibrium of the bilinear approximation of the game objective,

$$\theta^1 \leftarrow \theta^1 + \underset{\Delta\theta^1 : \Delta\theta^1 + \theta^1 \in \Theta^1}{\operatorname{argmax}} \Delta\theta^{1\top} D_{\theta^1}\eta + \Delta\theta^{1\top} D_{\theta^1\theta^2}\eta \Delta\theta^2 - \frac{1}{2\alpha}\|\Delta\theta^1\|^2,$$

$$\theta^2 \leftarrow \theta^2 + \underset{\Delta\theta^2 : \Delta\theta^2 + \theta^2 \in \Theta^2}{\operatorname{argmin}} \Delta\theta^{2\top} D_{\theta^2}\eta + \Delta\theta^{2\top} D_{\theta^2\theta^1}\eta \Delta\theta^1 + \frac{1}{2\alpha}\|\Delta\theta^2\|^2, \quad (5)$$

which has an extra term, the interaction term, in contrast to Eq. 4, and has the following closed-form solution,

$$\theta^1 \leftarrow \theta^1 + \alpha\big(I + \alpha^2 D_{\theta^1\theta^2}\eta D_{\theta^2\theta^1}\eta\big)^{-1}\big(D_{\theta^1}\eta - \alpha D_{\theta^1\theta^2}\eta D_{\theta^2}\eta\big),$$
$$\theta^2 \leftarrow \theta^2 - \alpha\big(I + \alpha^2 D_{\theta^2\theta^1}\eta D_{\theta^1\theta^2}\eta\big)^{-1}\big(D_{\theta^2}\eta + \alpha D_{\theta^2\theta^1}\eta D_{\theta^1}\eta\big), \quad (6)$$



(a) (b)

Figure 1: Bilinear games: (a) CoPO updates towards Nash equilibrium. (b) GDA updates point outward, leading to cycling/divergence.

where I is an identity matrix of appropriate size. Note that, the bilinear approximation in Eq. 5 is still linear in each player's action/parameters. Including any other terms, e.g., block diagonal Hessian terms from the Taylor expansion of the game objective, results in nonlinear terms in at least one player's parameters. As such, CoPO can be viewed as a natural linear generalization of PG. It is known that GDA-style co-learning approaches can often diverge or cycle indefinitely and never converge [Mertikopoulos et al., 2018a]. Fig. 1 shows that for a bilinear game, the gradient flow of GDA cycles or has gradient flow outward, while the CoPG flow, considering players' future moves, has gradient flow toward the Nash equilibrium and converges. In Apx.9, we deploy the Neumann series expansion of the inverses in Eq.5, and show that CoPO recovers the infinite recursion reasoning between players and the environment, while GDA correspond to the first term, and LOLA corresponds to the first two terms in the series. Next, we compute terms in Eq. 6 using the score function $g(\tau, \theta^i) := D_{\theta^i}(\log \prod_{k=0}^{|\tau|-1} \pi(a_k|s_k;\theta^i))$,

Proposition 1. *Given a* CoMDP*, players* $i, j \in \{1, 2\}$, $i \neq j$ *and the policy parameters* θ^i, θ^j,

$$D_{\theta^i}\eta = \int_\tau f(\tau;\theta^1,\theta^2) g(\tau,\theta^i) \mathsf{R}(\tau) d\tau,$$
$$D_{\theta^i\theta^j}\eta = \int_\tau f(\tau;\theta^1,\theta^2) g(\tau,\theta^i) g(\tau,\theta^j)^\top \mathsf{R}(\tau) d\tau.$$

Proof in Apx. 10.1. In practice, we use conjugate gradient and Hessian vector product to efficiently compute the updates in Eq.6, as explained in later sections. A closer look at CoPO shows that this paradigm does not require the knowledge of the environment if sampled trajectories are available. It neither requires full observability of the states, nor any structural assumption on CoMDP, but the Monte Carlo estimation suffer from high variance. In the following, we explicitly take the CoMDP structure into account to develop efficient algorithms.

3.1 COMPETITIVE POLICY GRADIENT

We propose competitive policy gradient (CoPG), an efficient algorithm that exploits the structure of CoMDPs to compute the parameter updates. The following is the CoMDP generalizing of the single agent PG theorem [Sutton et al., 2000]. For $\tau_{l:l'} = (s_k, a_k^1, a_k^2, r_k)_{k=l}^{l'}$, the events from time step l to l', we have:

Theorem 1. *Given a* CoMDP*, players* $i, j \in \{1, 2\}$, $i \neq j$, *and the policy parameters* θ^i, θ^j,

$$D_{\theta^i}\eta = \int_\tau \sum_{k=0}^{|\tau|-1} \gamma^k f(\tau_{0:k};\theta^1,\theta^2) D_{\theta^i}(\log \pi(a_k^i|s_k;\theta^i)) Q(s_k, a_k^1, a_k^2;\theta^1,\theta^2) d\tau, \quad (7)$$

$$D_{\theta^i\theta^j}\eta = \mathscr{T}_1 + \mathscr{T}_2 + \mathscr{T}_3. \quad (8)$$

Proof in Apx. 10.2. $\mathscr{T}_1$, $\mathscr{T}_2$ and $\mathscr{T}_3$ are described in Table 1. This theorem indicates that the terms in Eq. 6 can be computed using Q function. In Eq. 8, $\mathscr{T}_1$ is the immediate interaction between players, $\mathscr{T}_2$ is the interaction of player i's behavior up to time step k with player j's reaction at time step k, and the environment. $\mathscr{T}_3$ is the interaction of player j's behavior upto time step k with player i's reaction at time step k, and the environment.

CoPG operates in epochs. At each epoch, CoPG deploys $\pi(\theta^1), \pi(\theta^2)$ to collect trajectories, exploits them to estimate the Q, $D_{\theta^i}\eta$, and $D_{\theta^i\theta^j}\eta$. Then CoPG follows the parameter updates in Eq. 6 and updates (θ^1, θ^2), and this process goes on to the next epoch (Alg.2). Many variants of PG approach uses baselines, and replace Q with, the advantage function $A(s, a^1, a^2;\theta^1,\theta^2) = Q(s, a^1, a^2;\theta^1,\theta^2) - V(s;\theta^1,\theta^2)$, Monte Carlo estimate of Q-V [Baird, 1993], empirical TD error or generalized advantage function (GAE) [Schulman et al., 2016]. Our algorithm can be extended to this set up and in Apx. 11 we show how CoPG can be accompanied with all the mentioned variants.

Table 1: Notations for the bilinear term in the competitive policy theorem

Symbol	Formulation
$\mathscr{T}_1:$	$\int_\tau \sum_{k=0}^{\lvert\tau\rvert-1} \gamma^k f(\tau_{0:k};\theta^1,\theta^2) D_{\theta^i}(\log \pi(a_k^i\lvert s_k;\theta^i)) D_{\theta^j}(\log \pi(a_k^j\lvert s_k;\theta^j))^\top Q(s_k,a_k^1,a_k^2;\theta^1,\theta^2) d\tau$
$\mathscr{T}_2:$	$\int_\tau \sum_{k=1}^{\lvert\tau\rvert-1} \gamma^k f(\tau_{0:k};\theta^1,\theta^2) D_{\theta^i}(\log \prod_{l=0}^{k-1} \pi(a_l^i\lvert s_l;\theta^i)) D_{\theta^j}(\log \pi(a_k^j\lvert s_k;\theta^j))^\top Q(s_k,a_k^1,a_k^2;\theta^1,\theta^2) d\tau$
$\mathscr{T}_3:$	$\int_\tau \sum_{k=1}^{\lvert\tau\rvert-1} \gamma^k f(\tau_{0:k};\theta^1,\theta^2) D_{\theta^i}(\log \pi(a_k^i\lvert s_k;\theta^i)) D_{\theta^j}(\log \prod_{l=0}^{k-1} \pi(a_l^j\lvert s_l;\theta^j))^\top Q(s_k,a_k^1,a_k^2;\theta^1,\theta^2) d\tau$

3.2 OPPONENT PARAMETER ESTIMATION

In some settings, each learner does not have access to the opponent's policy. To apply CoPG in such settings, one natural approach is to estimate online the opponent's policy by the opponent's state-action pairs, as proposed in [Foerster et al., 2017b]. We thus propose a variant of CoPG that also infers the opponent's policy parameters (CoPG-OP), where each agent i also estimates the parameters $\hat{\theta}^j$ of its opponent j's policy, e.g., using maximum-likelihood estimator,

$$\hat{\theta}^j = \text{argmax}_{\theta^j} \mathbb{E}_{\tau\sim f(\cdot,\theta^1,\theta^2)} \sum_{k=0}^{|\tau|-1} \log \pi(a_k^j|s_k,\theta^j) \quad (9)$$

Then, the agent utilizes $\hat{\theta}^j$ to derive its policy updates in Eq. 7, 8, in place of θ^j. In our empirical study, we observe that CoPG-OP training is as stable as CoPG and the policies learned using CoPG-OP are as competent as CoPG (refer to Apx. 16). We conclude that opponent parameter learning can be considered effective in online settings, which confirms the observation in [Foerster et al., 2017b].

3.3 TRUST REGION COMPETITIVE POLICY OPTIMIZATION

Trust region based policy optimization methods exploit the local Riemannian geometry of the parameter space to derive more efficient policy updates [Kakade and Langford, 2002, Kakade, 2002, Schulman et al., 2015]. In this section, we propose trust region competitive policy optimization (TRCoPO), the CoPO generalization of TRPO [Schulman et al., 2015], that exploits the local geometry of the competitive objective to derive more efficient parameter updates.

Lemma 1. *Given the advantage function of policies $\pi(\theta^1), \pi(\theta^2)$, $\forall(\theta'^1,\theta'^2) \in \Theta^1\times\Theta^2$ we have,*

$$\eta(\theta'^1,\theta'^2) = \eta(\theta^1,\theta^2) + \mathbb{E}_{\tau\sim f(\cdot;\theta'^1,\theta'^2)} \sum_{k=0}^{|\tau|-1} \gamma^k A(s,a^1,a^2;\theta^1,\theta^2). \quad (10)$$

Proof in Apx. 13.1. Eq.10 indicates that, considering our current policies $(\pi(\theta^1),\pi(\theta^2))$, having access to their advantage function, and also samples from $f(\cdot;\theta'^1,\theta'^2)$ of any (θ'^1,θ'^2) (without rewards), we can directly compute $\eta(\theta'^1,\theta'^2)$ and optimize for (θ'^1,θ'^2). However, in practice, we might not have access to $f(\cdot;\theta'^1,\theta'^2)$ for all (θ'^1,θ'^2) to accomplish the optimization task, therefore, direct use of Eq.10 is not favorable. Instead, we define a surrogate objective function, $L_{\theta^1,\theta^2}(\theta'^1,\theta'^2)$,

$$L_{\theta^1,\theta^2}(\theta'^1,\theta'^2) = \eta(\theta^1,\theta^2) + \mathbb{E}_{\tau\sim f(\cdot;\theta^1,\theta^2)}\Big[\sum_{k=0}^{|\tau|-1}\gamma^k \mathbb{E}_{\pi(a_k'^1|s_k;\theta'^1),\pi(a_k'^2|s_k;\theta'^2)} A(s_k,a_k'^1,a_k'^2;\theta^1,\theta^2)\Big], \quad (11)$$

which can be computed using trajectories of our current polices $\pi(\theta^1),\pi(\theta^2)$. $L_{\theta^1,\theta^2}(\theta'^1,\theta'^2)$ is an object of interest in PG [Kakade and Langford, 2002, Schulman et al., 2015] since mainly its gradient is equal to that of $\eta(\theta^1,\theta^2)$ at (θ^1,θ^2), and as stated in the following theorem, it can carefully be used as a surrogate of the game value. For KL divergence $D_{KL}((\theta^1,\theta^2),(\theta'^1,\theta'^2)) := \int_\tau f(\tau,\theta^1,\theta^2)\log\big(f(\tau,\theta^1,\theta^2)/f(\tau,\theta'^1,\theta'^2)\big)d\tau$, we have,

Theorem 2. *$L_{\theta^1,\theta^2}(\theta'^1,\theta'^2)$ approximate $\eta(\theta'^1,\theta'^2)$ up to the following error upper bound, with constant ϵ*

$$\big|\eta(\theta'^1,\theta'^2) - L_{\theta^1\theta^2}(\theta'^1,\theta'^2)\big| \le \epsilon/\sqrt{2}\sqrt{D_{KL}((\theta'^1,\theta'^2),(\theta^1,\theta^2))}, \quad (12)$$

$$\epsilon := \max_\tau \sum_k^{|\tau|} \gamma^k \mathbb{E}_{\pi(a_k'^1|s_k;\theta'^1),\pi(a_k'^2|s_k;\theta'^2)} A(s_k,a_k'^1,a_k'^2;\theta^1,\theta^2).$$

Proof in Apx. 13.2. This theorem states that we can use $L_{\theta^1\theta^2}(\theta'^1,\theta'^2)$ to optimize for $\eta(\theta'^1,\theta'^2)$ as long as we keep the (θ'^1,θ'^2) in the vicinity of θ^1,θ^2. Similar to single agent TRPO [Schulman et al., 2015], we optimize for $L_{\theta^1\theta^2}(\theta'^1,\theta'^2)$, while constraining the KL divergence, $D_{KL}((\theta^1,\theta^2),(\theta'^1,\theta'^2)) \le \delta'$, i.e.,

$$\max_{\theta'^1\in\Theta^1}\min_{\theta'^2\in\Theta^2} L_{\theta^1\theta^2}(\theta'^1,\theta'^2), \text{with } D_{KL}((\theta^1,\theta^2),(\theta'^1,\theta'^2)) \le \delta'. \quad (13)$$

The GDA generalizing of TRPO uses a linear approximation of $L_{\theta^1\theta^2}(\theta'^1,\theta'^2)$ to approach this optimization, which again ignores the players' interactions. In contrast, TRCoPO considers the game theoretic nature of this optimization, and uses a bilinear approximation. TRCoPO operates in epochs. At each epoch, TRCoPO deploys $(\pi(\theta^1),\pi(\theta^2))$ to collect trajectories, exploits them to estimate A (or GAE), then updates parameters accordingly, Alg.4. (For more details, please refer to Apx. 13.3.)

4 EXPERIMENTS

We empirically study the performance of CoPG and TRCoPO and their counterparts GDA and TRGDA, on six games, ranging from single-state repeated games to general sequential games, and tabular games to infinite/continuous high dimensional states/action games. They are 1) linear-quadratic(LQ) game, 2) bilinear game, 3) matching pennies (MP), 4) rock paper scissors (RPS), 5) Markov soccer, and 6) car racing. These games are representative enough that their study provides insightful conclusions, and challenging enough to highlight the core difficulties and interactions in competitive games.

We show that CoPG and TRCoPO converge to stable points, and learn opponent aware strategies, whereas GDA's and TRGDA's greedy approach shows poor performance and even diverge in bi-linear, MP, and RPS games. For the LQ game, when GDA does not diverge, it almost requires 1.5 times the amount of samples, and is 1.5 times slower than CoPG. In highly strategic games, where players' policies are tightly coupled, we show that CoPG and TRCoPO learn much better interactive strategies. In the soccer game, CoPG and TRCoPO players learn to defend, dodge and score goals, whereas GDA and TRGDA players learn how to score when they are initialized with the ball, and give way to the other player otherwise. In the car racing, while GDA and TRGDA show poor performance, CoPG and TRCoPO produce competing players, which learn to block and fake each other. Overall, we observe CoPG and TRCoPO considerably outperform their counterparts in terms of convergence, learned strategies, and gradient dynamics.

We implemented all algorithms in Pytorch [Paszke et al., 2019], and made the code and the videos public[1]. In our implementation, we deploy Pytorch's autograd package and Hessian vector product to efficiently obtain gradients and Hessian vector products to compute the bilinear terms in the optimizer. Moreover, we use the conjugate gradient trick to efficiently computed the inverses-matrix vector product in Eq.6 which incurs a minimal computational overhead (see [Shewchuk, 1994] for more details). To improve computation times, we compute inverse-matrix vector product only for one player strategy, and use optimal counter strategy for other player $\Delta\theta^2$ which is computed without an inverse matrix vector product. Also, the last optimal strategy can be used to warm start the conjugate gradient method which improves convergence times. We provide efficient implementation for both CoPO- and GDA-based methods, where CoPO incurs 1.5 times extra computation per batch.

Zero-sum LQ game is a continuous state-action linear dynamical game between two players, where GDA, with considerably small learning rate, has favorable convergence guarantees [Zhang et al., 2019]. This makes the LQ game an ideal platform to study the range of allowable step sizes and convergence rate of CoPG and GDA. We show that, with increasing learning rate, GDA generates erratic trajectories and policy updates, which cause instability (see "⊘" in Table 4), whereas CoPG is robust towards this behavior. Fig. 2a shows that CoPG dynamics are not just faster at the same learning rate but more importantly, CoPG can potentially take even larger steps.

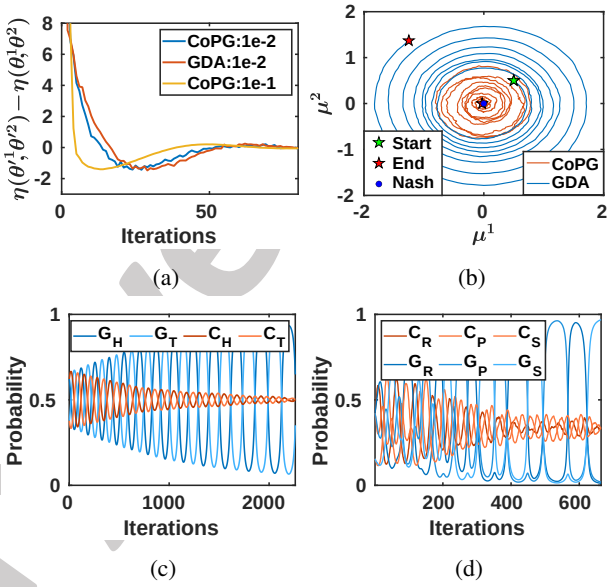


Figure 2: During training CoPG (C) vs GDA (G) on a) LQ game, difference in game objective due to policy update for $\alpha = 1e-1, 1e-2$ b) Bilinear game, μ^1 vs μ^2 c) MP, probability of selecting Head(H) and Tail(T) d) RPS, probability of selecting rock(R), paper(P) and scissors(S).

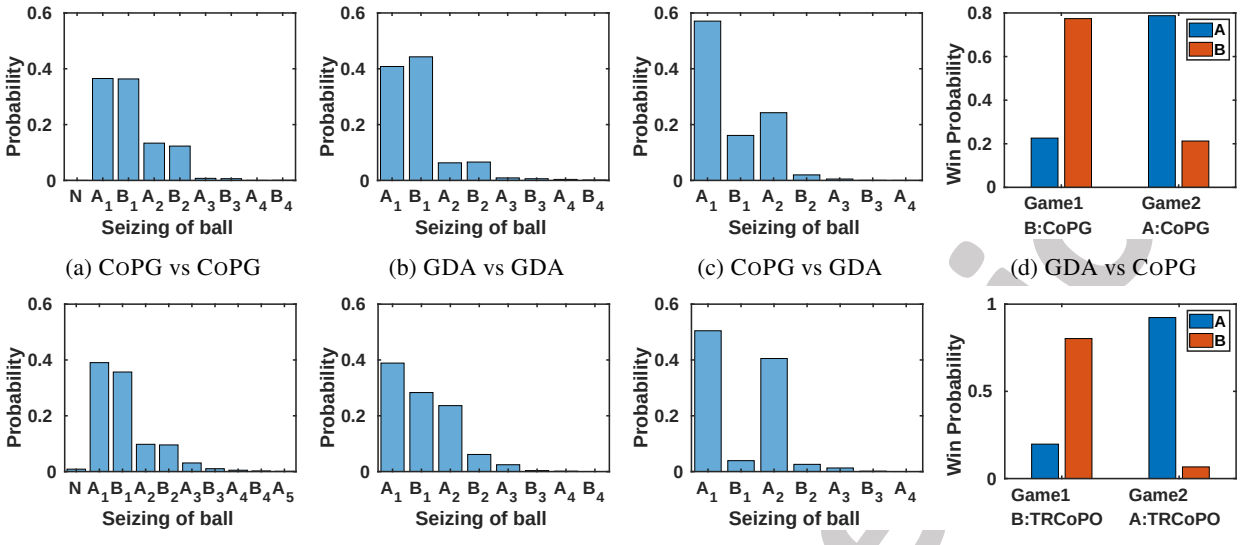
Bilinear game is a state-less strongly non-cooperative game, where GDA is known to diverge [Balduzzi et al., 2018]. In this game, reward $r(a^1, a^2) = \langle a^1, a^2 \rangle$ where $a^1 \sim \mathcal{N}(\mu_1, \sigma_1)$ and $a^2 \sim \mathcal{N}(\mu_2, \sigma_2)$, and (μ_1, σ_1), (μ_2, σ_2) are policy parameters. We show that GDA diverges for all learning rates, whereas CoPG converges to the unique Nash equilibrium Fig. 2b.

Matching pennies and **Rock paper scissors**, are challenging matrix games with mixed strategies as Nash equilibria, demand opponent aware optimization.[2] We show that CoPG and TRCoPO both converge to the unique Nash equilibrium of MP Fig. 2c and RPS Fig. 2d, whereas GDA and TRGDA diverges (to a sequence of polices that are exploitable by deterministic strategy). Detailed empirical study, formulation and explanation of these 4 games can be found in Apx. 15.

Markov soccer game, Fig. 4, consists of players A and B that are randomly initialised in the field, that are supposed to pick up the ball and put it in the opponent's goal [Littman,

[1]Link to Videos: https://sites.google.com/view/rl-copo

[2]Traditionally, fictitious and counterfactual regret minimization approaches have been deployed [Neller and Lanctot, 2013].



(a) CoPG vs CoPG (b) GDA vs GDA (c) CoPG vs GDA (d) GDA vs CoPG

(e) TRCoPO vs TRCoPO (f) TRGDA vs TRGDA (g) TRCoPO vs TRGDA (h) TRGDA vs TRCoPO

Figure 3: a-c), e-g) Interaction plots, representing probability of seizing ball in the game between A vs B. X-axis convention, for player A. A_1: A scored goal, A_2: A scored goal after seizing ball from B, A_3: A scored goal by seizing ball from B which took the ball from A and so forth. Vice versa for player B. N: No one scored goal both kept on seizing ball. d),h) Probability of games won.

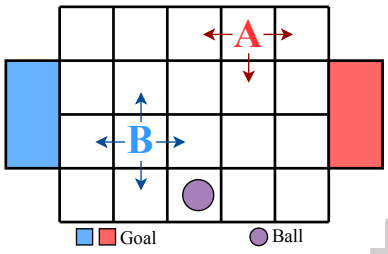


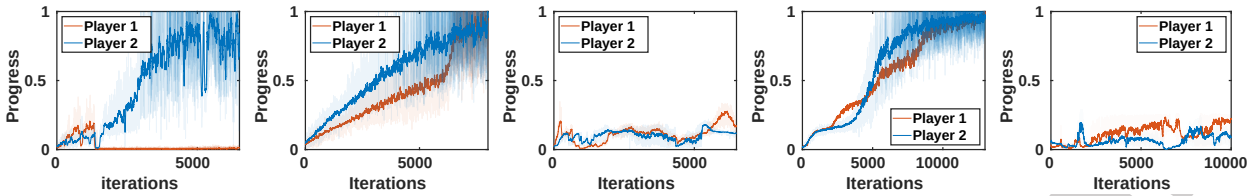
Figure 4: Markov Soccer

1994, He et al., 2016]. The winner is awarded with +1 and the loser with -1 (see Apx. 15.5 for more details).

Since all methods converge in this game, it is a suitable game to compare learned strategies. In this game, we expect a reasonable player to learn sophisticated strategies of defending, dodging and scoring. For each method playing against their counterparts, e.g., CoPG against CoPG and GDA, Fig. 3 shows the number of times the ball is seized between the players before one player finally scores a goal, or time-out. In (3a), CoPG vs CoPG, we see agents seize ball, twice the times as compared to (3b),GDA vs GDA (see A_2 and B_2 in (3a) and (3b)). In the matches CoPG vs GDA (3c), CoPG trained agent could seize the ball from GDA agent (A_2) nearly 12 times more due to better seizing and defending strategy, but GDA can hardly take the ball back from CoPG (B_2) due to a better dodging strategy of the CoPG agent. Playing CoPG agent against GDA one, we observe that CoPG wins more than 74% of the games Fig. 3d. We observe a similar trend for trust region based methods TRCoPO and TRGDA, playing against each other (a slightly stronger results of 85% wins, A_2 column in Figs 3g, 3c).

We also compared CoPG with MADDPG [Lowe et al., 2017] and LOLA [Foerster et al., 2017b]. We observe that the MADDPG learned policy behaves similar to GDA, and loses 80% of the games to CoPG's (Fig. 15b). LOLA learned policy, with its second level reasoning, performs better than GDA, but lose to CoPG 72% of the matches. For completeness, we also compared GDA-PG, CoPG, TRGDA, and TRCoPO playing against each other. TRCoPO performs best,CoPG was runner up, then TRGDA, and lastly GDA (see Apx. 15.5).

Car Racing is another interesting game, with continuous state-action space, where two race cars competing against each other to finish the race first [Liniger and Lygeros, 2020]. The game is accompanied by two important challenges, 1) learning a policy that can maneuver the car at the limit of handling, 2) strategic interactions with opponents. The track is challenging, consisting of 13 turns with different curvature (Fig. 6). The game is formulated as a zero-sum, with reward $r(s_k, a_k^1, a_k^2) = \Delta\rho_{car_1} - \Delta\rho_{car_2}$, where $\Delta\rho = \rho_{k+1} - \rho_k$ and ρ_k is the car's progress along the track at the k^{th} time step. If a car crosses track boundaries (e.g., hit the wall), it is penalized, and the opponent receives rewards, this encourages cars to play rough and push each other into the track boundaries. When a collision happens, the rear car is penalized, and the car in the front receives a reward; it promotes blocking by the car in front and overtaking by the car in the rear. We study agents trained with all GDA, TRGDA, MADDPG, LOLA, CoPG and TRCoPO in this game, and show that even though all players were able to learn to "drive" only CoPG and TRCoPO were able to learn how to "race". Using GDA, only one player was able to learn, which manifested in either one player



(a) GDA (b) CoPG (c) TRGDA (d) TRCoPO (e) MADDPG

Figure 5: Normalised progress of agents in one lap of car racing game during training

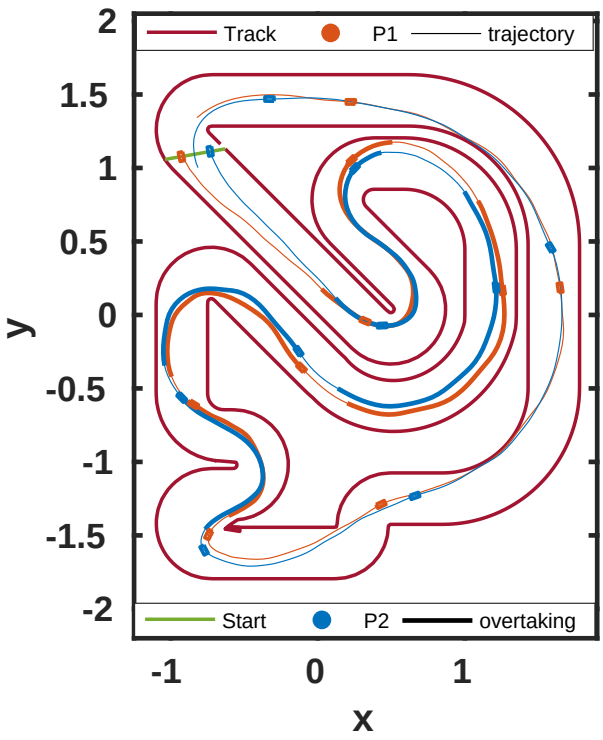


Figure 6: Overtaking maneuvers of CoPO agents in the Car Racing Game. The thin line shows the trajectory of the player in the game and the thick line shows trajectory when the trailing agent overtook.

completely failing and the other finishing the track (Fig. 5a), or by an oscillation behavior where one player learns to go ahead, the other agent stays at lower progress(Fig. 5c, Fig. 5e). Even if one of the players learns to finish one lap at some point, this player does not learn to interact with its opponent(https://youtu.be/rxkGW02GwvE). In contrast, players trained with CoPG and TRGDA, both progress, learn to finish the lap, and race (interact with each other) (See Fig. 5b and Fig. 5d). To test the policies, we performed races between CoPG and GDA, TRCoPO and TRGDA, and CoPG and TRCoPO. As shown in Table 2, CoPG and TRCoPO win almost all races against their counterparts. Overall, we see that both CoPG and TRCoPO are able to learn policies that are faster, more precise, and interactive with the other player (e.g., learns to overtake).

Table 2: Trained agents competing against each others in car racing. Ratio of Wins(W), Overtakes(O) and Collisions(C).

	CoPG v GDA		TRCoPO v TRGDA		CoPG v TRCoPO	
W	**1**	0	**1**	0	0.24	**0.76**
O	**1.28**	0.78	**1.28**	0.78	1.80	**2.07**
C	**0.17**	16.11	**0.25**	1.87	**0.30**	0.31

4.1 CASE STUDIES

Generative Adversarial Imitation Learning: One can also apply our approach in asymmetric games, such as learning the agent policy and the discriminator in generative adversarial imitation learning (GAIL) [Ho and Ermon, 2016]. In GAIL, to imitate the expert, the agent (θ player) plays a game with a discriminator $\mathbb{D}$ (ϕ player), i.e.,

$$\min_{\theta}\max_{\phi}\mathbb{E}_{\tau_\theta}[\log\mathbb{D}_\phi(s,a)]+\mathbb{E}_{\tau_e}[\log(1-\mathbb{D}_\phi(s,a))]-\lambda H(\theta),$$

where τ_θ is a trajectory and $H(\theta)$ is the casual entropy.

We conduct this study on the car racing game with a single car, where the aim is to learn to drive a full lap. Given a long track (Fig. 6), exploration and reward formulation can be challenging. We train the agent using CoPO to learn to drive by imitating an expert. The expert trajectories are collected using a pure pursuit (PP) controller [Coulter, 1992] with different sets of parameters Apx. 17. We evaluate agent's policy using lap time (t^{lap}), and:

$$\zeta = E_{\tau_\theta}[\sum_{k}^{|\tau|-1}\|a_k - a_k^e\|^2/|\tau|], \qquad (14)$$

where a_k, a_k^e are the agent and the expert actions evaluated at the same state s_k, collected from an agent rollout. We compare the results of CoPO-GAIL with GDA-GAIL, Behaviour cloning (BC) [Ho and Ermon, 2016] and controllers such as PID and PP. The agent trained with CoPO learns to follow the reference path of the expert and even drives better than the average expert policy, achieving a performance similar to the best expert (https://youtu.be/DtGWZubjcf4). The CoPO-based agent achieves the lowest ζ value and learns a better imitation policy compared to GDA and BC Table 3.

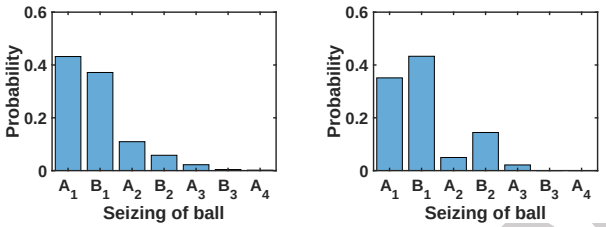
Opponent learning: We next explore the case where one does not directly have access to the opponent's parameters

Table 3: Comparing lap time and ζ (Eq. 14) of the policies learnt using imitation learning and the baseline controllers.

score	PP_{avg}	CoPO-GAIL	GDA-GAIL	PID	BC
ζ x 10^{-2}	0	0.82	1.14	2.35	2.02
$t^{lap}(s)$	13.07	11.82	12.18	14.67	DNF

and they have to be inferred through interaction with the other agent and the environment. We propose to use CoPG-OP (Sec. 3.2), the opponent learning variant of CoPG.

Fig. 7 shows interaction plots of CoPG-OP conducted on the Markov Soccer game (setting explained in Apx. 15.5). The opponent's parameters are estimated by observing state-action pairs of the opponent using Eq. 9. The interaction plot of the CoPG-OP agent with the estimated opponent (Fig. 7a) shows that the CoPG-OP player learns to seize the ball and interact with the opponent(A_3, A_4). Fig. 18b shows the interaction plot of CoPG-OP with CoPG, where we observe that CoPG-OP also learns a policy similar to CoPG, which is able to defend, escape and score goals (A_3). When directly competing with CoPG, we observe that CoPG-OP can win 46.5% of the games against CoPG.



(a) CoPG-OP (A) vs (B Est A) (b) CoPG-OP vs CoPG

Figure 7: Interaction plots evaluated by playing 5000 games. Matches played between a) CoPG-OP player A and player B estimated by A b) CoPG-OP and CoPG

The experiment details and numerical results on other setups can be found in Apx. 16. They show that CoPG-OP achieves the performance of CoPG in terms of stability and convergence to sophisticated strategies.

Training by Self-play: In self-play, one player plays against itself using the same policy model for both players. Each player then samples actions from this policy for their respective state and updates the policy using CoPG-SP (Alg. 5) which is a special case of CoPG. We observe that CoPG-SP can successfully learn competing strategies similarly to CoPG. We provide the CoPG-SP algorithm and a detailed empirical study in Apx. 12.

5 RELATED WORK

In tabular CoMDP, Q-learning and actor-critic have been deployed [Littman, 1994, 2001b,a, Bowling and Veloso, 2002, Greenwald and Hall, 2003, Hu and Wellman, 2003, Frénay and Saerens, 2009, Pérolat et al., 2018, Srinivasan et al., 2018], and recently, deep RL methods have been extending to CoMDPs, with focus on modeling agents behaviour [Tampuu et al., 2017, Leibo et al., 2017, Raghu et al., 2017]. To mitigate the stabilization issues, centralized methods [Matignon et al., 2012, Lowe et al., 2017, Foerster et al., 2017a], along with opponent's behavior modeling [Raileanu et al., 2018, He et al., 2016] have been explored. Optimization in multi-agent learning can be interpreted as a game in the parameter space, and the main body of the mentioned literature does not take this aspect directly into account since they attempt to separately improve players' performance. Hence, they often fail to achieve desirable performance and oftentimes suffer from unstable training, especially in strategic games [Hernandez-Leal et al., 2019, Buşoniu et al., 2010]. In imperfect information games with known rules, e.g., poker [Moravcík et al., 2017], a series of works study algorithmically computing Nash equilibra [Shalev-shwartz and Singer, 2007, Koller et al., 1995, Gilpin et al., 2007, Zinkevich et al., 2008, Bowling et al., 2017]. Also, studies in stateless episodic games shown convergence to coarse correlated equilibrium [Hartline et al., 2015, Blum et al., 2008]. In contrast CoPO converges to the Nash equilibrium in such games. In two-player competitive games, self-play is an approach of interest where a player plays against itself to improve its behavior [Tesauro, 1995, Silver et al., 2016]. But, many of these approaches are limited to specific domains [Heinrich et al., 2015, Heinrich and Silver, 2016].

The closest approach to CoPO in the literature is LOLA [Foerster et al., 2017b] an opponent aware approach. LOLA updates parameters using a second-order correction term, resulting in gradient updates corresponding to the following shortened recursion: if a player thinks that the other player thinks its strategy stays constant [Schaefer and Anandkumar, 2019], whereas CoPO recovers the full recursion series until the Nash equilibrium is delivered. In contrast to [Foerster et al., 2017b] we also provide CoPO extension to value-based, and trust region-based methods, along with their efficient implementation.

Our work is also related to GANs [Goodfellow et al., 2014], which involves solving a zero sum two-player competitive game (CoMDP with single state). Recent development in nonconvex-nonconcave problems and GANs training show GDA has undesirable convergence properties [Mazumdar et al., 2019] and exhibit strong rotation around fixed points [Balduzzi et al., 2018]. To overcome this rotation behaviour of GDA, various modifications have been proposed, including averaging [Yazıcı et al., 2019], negative momentum [Gidel et al., 2018] along many others [Mertikopoulos et al., 2018b, Daskalakis et al., 2017, Mescheder et al., 2017, Balduzzi et al., 2018, Gemp and Mahadevan, 2018]. Considering the game-theoretic nature of this problem, competitive gradient descent has been proposed as a natural generalization of gradient descent in two-players instead of GDA for

GANs [Schaefer and Anandkumar, 2019]. This method, as the predecessor to CoPO, enjoys stability in training, robustness in choice of hyper-parameters, and has desirable performance and convergence properties.

6 DISCUSSION ON APPLICABILITY

CoPO is the paradigm of competitive policy optimization where the goal is to *jointly* find policies for agents. In CoPO, the optimization is centralized, and the execution of actions is decentralized. Applications of such setting are; *(i)* self-play: we train an agent to play against itself; *(ii)* adversarial robustness, inverse RL, and imitation learning: we aim to find a robust model; *(iii)* robust control [Zhou et al., 1996]: we train agents to be robust against attackers; *(iv)* athletic games analysis, e.g., soccer and basketball: we train models of teams in simulation, and let them play against each other to discover tactics and strategies; *(v)* Robocup World (robots soccer): we train our team in our lab before deploying it in the real match; *(vi)* AI economist [Zheng et al., 2020]: we run a game between workers along with rule-makers to discover new tax laws, and many more real-world problems.

Our empirical study shows that CoPG and TRCoPO can excel in this setting and have clear advantages compared to existing algorithms. While the centralized optimization setting in CoPO has a vast range of real-world applications, there are problems that require decentralized optimization. We showed that CoPG-OP, a decentralized extension of CoPG, where each agent also learns its opponent's parameters/model to compute its policy update, comes with the same benefits as CoPG in the centralized setting.

7 CONCLUSION

We presented competitive policy optimization CoPO, a novel PG-based RL method for two player strictly competitive game. In CoPO, each player optimizes strategy by considering the interaction with the environment and the opponent through game theoretic bilinear approximation to the game objective. This method is instantiated to competitive policy gradient (CoPG) and trust region competitive policy optimisation (TRCoPO) using value based and trust region approaches. We theoretically studied these methods and provided PG theorems to show the properties of these model-free RL approaches. We provided efficient implementation of these methods and empirically showed that they provide stable and faster optimization, and also converge to more sophisticated and competitive strategies. We performed case studies for CoPO based approach on self-play, asymmetric mini-max game GAIL, with opponent modelling and further discussed the general applicability of the CoPO paradigm in various real life settings. We dedicated this paper to two player zero-sum games, however, the principles provided in this paper can be used for multi-player general games. In the future, we plan to extend this study to multi-player general-sum games along with efficient implementation of methods. Moreover, we plan to use the techniques proposed in partially observable domains, and study imperfect information games [Azizzadenesheli et al., 2020].

Acknowledgements

The main body of this work took place when M. Prajapat was a visiting scholar at Caltech. The authors would like to thank Florian Schäfer for his support. M. Prajapat is thankful to Zeno Karl Schindler foundation for providing him with a Master thesis grant. K. Azizzadenesheli is supported in part by Raytheon. A. Anandkumar is supported in part by Bren endowed chair, DARPA PAIHR00111890035 and LwLL grants, Raytheon, Microsoft, Google, and Adobe faculty fellowships.

References

V. M. Aleksandrov, V. I. Sysoyev, and V. V. Shemeneva. Stochastic optimization. *Engineering Cybernetics*, 1968.

Kamyar Azizzadenesheli, Yisong Yue, and Anima Anandkumar. Policy gradient in partially observable environments: Approximation and convergence. *arXiv:1810.07900*, 2020.

Leemon C Baird, III. Advantage updating. *WRIGHT LAB WRIGHT-PATTERSON AFB OH*, 1993.

Egbert Bakker, Lars Nyborg, and Hans B. Pacejka. Tyre modelling for use in vehicle dynamics studies. In *SAE Technical Paper*. SAE International, 02 1987. doi: 10.4271/870421.

David Balduzzi et al. The mechanics of n-player differentiable games. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 354–363, 10–15 Jul 2018.

Avrim Blum et al. Regret minimization and the price of total anarchy. In *ACM Symposium on Theory of Computing*, page 373–382, 2008. ISBN 9781605580470.

Michael Bowling and Manuela Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136 (2):215–250, 2002. ISSN 0004-3702.

Michael Bowling et al. Heads-up limit hold'em poker is solved. *Commun. ACM*, 60(11):81–88, October 2017. ISSN 0001-0782. doi: 10.1145/3131284.

Lucian Buşoniu, Robertand Babuška, and Bart De Schutter. *Multi-agent Reinforcement Learning: An Overview*, pages 183–221. Springer Berlin Heidelberg, 2010.

R Craig Coulter. Implementation of the pure pursuit path tracking algorithm. Technical report, CMU RI, 1992.

Constantinos Daskalakis et al. Training gans with optimism. *CoRR*, abs/1711.00141, 2017.

Jerzy Filar and Koos Vrieze. *Competitive Markov decision processes*. Springer Science & Business Media, 2012.

Jakob N. Foerster et al. Counterfactual multi-agent policy gradients. *CoRR*, abs/1705.08926, 2017a.

Jakob N. Foerster et al. Learning with opponent-learning awareness. *CoRR*, abs/1709.04326, 2017b.

Benoít Frénay and Marco Saerens. Ql2, a simple reinforcement learning scheme for two-player zero-sum markov games. *Neurocomput.*, page 1494–1507, 2009.

Ian Gemp and Sridhar Mahadevan. Global convergence to the equilibrium of gans using variational inequalities. *CoRR*, abs/1808.01531, 2018.

Gauthier Gidel et al. Negative momentum for improved game dynamics. *CoRR*, abs/1807.04740, 2018.

Andrew Gilpin et al. Gradient-based algorithms for finding nash equilibria in extensive form games. In *WINE*, 2007.

Ian Goodfellow et al. Generative adversarial nets. In *Advances in neural information processing systems*, 2014.

Amy Greenwald and Keith Hall. Correlated-q learning. In *International Conference on Machine Learning*, 2003.

Jason Hartline, Vasilis Syrgkanis, and Éva Tardos. No-regret learning in bayesian games. In *Advances in Neural Information Processing Systems*, page 3061–3069, 2015.

He He, Jordan L. Boyd-Graber, Kevin Kwok, and Hal Daumé III. Opponent modeling in deep reinforcement learning. *CoRR*, abs/1609.05559, 2016.

Johannes Heinrich and David Silver. Deep reinforcement learning from self-play in imperfect-information games. *CoRR*, abs/1603.01121, 2016.

Johannes Heinrich, Marc Lanctot, and David Silver. Fictitious self-play in extensive-form games. In *Proceedings of the 32nd International Conference on Machine Learning - Volume 37*, ICML'15, page 805–813, 2015.

Pablo Hernandez-Leal, Bilal Kartal, and Matthew E. Taylor. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 33(6):750–797, Oct 2019. ISSN 1573-7454.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, 2016.

Junling Hu and Michael P. Wellman. Nash q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, page 1039–1069, 2003.

Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 2961–2970, 2019.

Sham Kakade. A natural policy gradient. In *Advances in neural information processing systems*, pages 1531–1538, 2002.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, pages 267–274, 2002.

John Maynard Keynes. *The general theory of employment, interest, and money*. Springer, 2018.

Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 1995.

Joel Z. Leibo et al. Multi-agent reinforcement learning in sequential social dilemmas. *CoRR*, abs/1702.03037, 2017.

Alexander Liniger and John Lygeros. A noncooperative game approach to autonomous racing. *IEEE Transactions on Control Systems Technology*, 28(3):884–897, 2020.

Alexander Liniger, Alexander Domahidi, and Manfred Morari. Optimization-based autonomous racing of 1: 43 scale rc cars. *Optimal Control Applications and Methods*, 36(5):628–647, 2015.

Michael Littman. Value-function reinforcement learning in markov games. *Cognitive Systems Research*, 2001a.

Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on International Conference on Machine Learning*, ICML'94, page 157–163, 1994. ISBN 1558603352.

Michael L. Littman. Friend-or-foe q-learning in general-sum games. In *International Conference on Machine Learning*, page 322–328, 2001b.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in Neural Information Processing Systems*, 2017.

Laetitia Matignon, Guillaume J. Laurent, and Nadine Le Fort-Piat. Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. *The Knowledge Engineering Review*, page 1–31, 2012.

Eric Mazumdar, Lillian J. Ratliff, Michael I. Jordan, and S. Shankar Sastry. Policy-gradient algorithms have no guarantees of convergence in linear quadratic games, 2019.

Panayotis Mertikopoulos, Christos Papadimitriou, and Georgios Piliouras. Cycles in adversarial regularized learning. In *ACM-SIAM Symposium on Discrete Algorithms*, 2018a.

Panayotis Mertikopoulos et al. Mirror descent in saddle-point problems: Going the extra (gradient) mile. *CoRR*, abs/1807.02629, 2018b.

Lars M. Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of gans. *CoRR*, abs/1705.10461, 2017.

Matej Moravcík et al. Deepstack: Expert-level artificial intelligence in no-limit poker. *CoRR*, abs/1701.01724, 2017.

Todd W. Neller and Marc Lanctot. An introduction to counterfactual regret minimization. *Educational Advances in Artificial Intelligence (EAAI-2013)*, 2013.

Adam Paszke et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019.

Julien Pérolat et al. Actor-critic fictitious play in simultaneous move multistage games. In *AISTATS*, volume 84 of *Proceedings of Machine Learning Research*, pages 919–928. PMLR, 2018.

Maithra Raghu et al. Can deep reinforcement learning solve erdos-selfridge-spencer games? *CoRR*, abs/1711.02301, 2017.

Roberta Raileanu et al. Modeling others using oneself in multi-agent reinforcement learning. *CoRR*, abs/1802.09640, 2018.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, 1951.

Florian Schaefer and Anima Anandkumar. Competitive gradient descent. In *Advances in Neural Information Processing Systems 32*, pages 7625–7635. Curran Associates, Inc., 2019.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *4th International Conference on Learning Representations, ICLR*, 2016.

John Schulman et al. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1889–1897, 07–09 Jul 2015.

Florian Schäfer, Hongkai Zheng, and Anima Anandkumar. Implicit competitive regularization in gans, 2019.

Shai Shalev-shwartz and Yoram Singer. Convex repeated games and fenchel duality. In *Advances in Neural Information Processing Systems*, pages 1265–1272. MIT Press, 2007.

Jonathan R Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, Carnegie Mellon University, USA, 1994.

David Silver et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

Sriram Srinivasan et al. Actor-critic policy optimization in partially observable multiagent environments. *CoRR*, abs/1810.09026, 2018.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Richard S Sutton, David A McAllester, et al. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

Ardi Tampuu et al. Multiagent cooperation and competition with deep reinforcement learning. *PLOS ONE*, 2017.

Gerald Tesauro. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

Jose L. Vázquez et al. Optimization-based hierarchical motion planning for autonomous racing. *ArXiv*, abs/2003.04882, 2020.

Yasin Yazıcı et al. The unusual effectiveness of averaging in GAN training. In *International Conference on Learning Representations*, 2019.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Policy optimization provably converges to nash equilibria in zero-sum linear quadratic games, 2019.

Stephan Zheng et al. The ai economist: Improving equality and productivity with ai-driven tax policies. *arXiv*, 2020.

Kemin Zhou et al. *Robust and optimal control*, volume 40. Prentice hall New Jersey, 1996.

Martin Zinkevich et al. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems 20*, pages 1729–1736. Curran Associates, Inc., 2008.