Competitive Policy Optimization

Manish Prajapat\textsuperscript{1,3} Kamyar Azizzadenesheli\textsuperscript{2} Alexander Liniger\textsuperscript{1} Yisong Yue\textsuperscript{3} Anima Anandkumar\textsuperscript{3}

\textsuperscript{1}ETH Zurich,\textsuperscript{2}Purdue University,\textsuperscript{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 sometimes 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
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 $\langle S, A1, A2, R, T, P, \gamma \rangle$, where $S$ is the state space, $s \in S$ is a state, for player $i \in \{1, 2\}$, $A_i$ is the player $i$’s action space with $a_i \in A_i$. $R$ is the reward kernel with probability distribution $R(s,a_1,a_2)$ and mean function $r(s,a_1,a_2)$ on $\mathbb{R}$. For a probability measure $P$, $p$ denotes the probability distribution of initial state, and for the transition kernel $T$, $T(s’,s,a_1,a_2)$ 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 $\Theta^i \subset \mathbb{R}^i$ 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 $\tau = ((s_0,a_1^0,a_2^0,r_0,k_{\tau})|_{k=0}^{\tau-1},s_{\tau})$, 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 $\tau$ following players’ policies $\pi(\theta^i)$.

$$f(\tau;\theta^1,\theta^2) = p(s_0) \prod_{k=0}^{\tau-1} \pi(a_1^k|s_k;\theta^1)\pi(a_2^k|s_k;\theta^2)$$

$$R(r_k|s_k,a_1^k,a_2^k)T(s_{k+1}|s_k,a_1^k,a_2^k).$$

For $R(\tau) = \sum_{k=0}^{\tau-1} \gamma^k r(s_k,a_1^k,a_2^k)$, the Q-function, V-functions, and game objective are defined,

$$Q(s_k,a_1^k,a_2^k;\theta^1,\theta^2) = E_{\tau \sim f(\cdot;\theta^1,\theta^2)} \left[ \sum_{j=k}^{\tau-1} \gamma^{j-k} r(s_j,a_1^j,a_2^j)|s_k,a_1^k,a_2^k) \right],$$

$$V(s_k;\theta^1) = E_{\tau \sim f(\cdot;\theta^1)} \left[ \sum_{j=k}^{\tau-1} \gamma^{j-k} r(s_j,a_1^j,a_2^j)|s_k) \right],$$

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

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

3 COMPETITIVE POLICY OPTIMIZATION

Player 1 aims to maximize the game objective $\eta$ 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,

$$\theta^1* \in \arg\max_{\theta^1 \in \Theta^1} \eta(\theta^1, \theta^2^2), \text{ and } \theta^2* \in \arg\min_{\theta^2 \in \Theta^2} \eta(\theta^1, \theta^2^2).$$

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

$$\theta^1 \leftarrow \theta^1 + \alpha \arg\max_{\Delta \theta^1: \Delta \theta^1 + \theta^1 \in \Theta^1} \Delta \theta^1^T D_{\theta^1} \eta - \frac{1}{2\alpha} ||\Delta \theta^1||^2,$$

$$\theta^2 \leftarrow \theta^2 + \alpha \arg\min_{\Delta \theta^2: \Delta \theta^2 + \theta^2 \Theta^2} \Delta \theta^2^T D_{\theta^2} \eta + \frac{1}{2\alpha} ||\Delta \theta^2||^2,$$

where $\alpha$ 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 [Schaef 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 [Schaef 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,

\[ \theta^1 \leftarrow \theta^1 + \arg \max_{\theta^1} \Delta \theta^1 D_{\theta^1 \theta^2} \eta^1 + \Delta \theta^1 D_{\theta^1 \theta^2} \eta^2 \Delta \theta^2 - \frac{1}{2\alpha} ||\Delta \theta^1||^2 \]

\[ \theta^2 \leftarrow \theta^2 + \arg \min_{\theta^2} \Delta \theta^2 D_{\theta^2 \theta^1} \eta^2 + \Delta \theta^2 D_{\theta^2 \theta^1} \eta^1 + \frac{1}{2\alpha} ||\Delta \theta^2||^2 \]

which has an extra term, the interaction term, in contrast to Eq. [5] and has the following closed-form solution,

\[ \theta^1 \leftarrow \theta^1 + (I + \alpha \Delta \theta^1 D_{\theta^1 \theta^2} \eta^2) (D_{\theta^1 \eta^2} - \alpha D_{\theta^2 \theta^1} \eta^2 D_{\theta^2 \eta^1}), \]

\[ \theta^2 \leftarrow \theta^2 - (I + \alpha \Delta \theta^2 D_{\theta^2 \theta^1} \eta^1) (D_{\theta^2 \eta^1} + \alpha D_{\theta^1 \theta^2} \eta^1 D_{\theta^1 \eta^2}). \]

![Figure 1](image-link) 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 CoPO flow, considering player’s 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 \( g(\tau, \theta^i) := D_{\theta^i} \left( \log \prod_{k=0}^{\tau-1} \pi(a_k | s_k; \theta^i) \right) \).

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

\[ D_{\theta^i \theta^j} \eta = \int_\tau f(\tau; \theta^i, \theta^j) g(\tau, \theta^i) R(\tau) d\tau, \]

\[ D_{\theta^j \theta^i} \eta = \int_\tau f(\tau; \theta^j, \theta^i) g(\tau, \theta^j) R(\tau) d\tau. \]

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 \( \tau_{l:l} = (s_k, a_k, r_k, e_k)_{k=l}^l \), 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 \( \theta^i, \theta^j \),

\[ D_{\theta^i \theta^j} \eta = \int_\tau \sum_{k=0}^{\tau-1} \gamma^k f(\tau_{k:k}; \theta^i, \theta^j) D_{\theta^i} \left( \log \pi(a_k | s_k; \theta^i) \right) Q(s_k, a_k, s_k; \theta^i \theta^j) d\tau, \]

\[ D_{\theta^j \theta^i} \eta = \int_\tau \sum_{k=0}^{\tau-1} \gamma^k f(\tau_{k:k}; \theta^j, \theta^i) D_{\theta^i} \left( \log \pi(a_k | s_k; \theta^i) \right) Q(s_k, a_k, s_k; \theta^i \theta^j) d\tau. \]

Proof in Apx[10.2] \( \mathcal{I}_1, \mathcal{I}_2 \) and \( \mathcal{I}_3 \) are described in Table[1] This theorem indicates that the terms in Eq[6] can be computed using Q function. In Eq[8] \( \mathcal{I}_1 \) is the immediate interaction between players, \( \mathcal{I}_2 \) 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. \( \mathcal{I}_3 \) is the interaction of player \( j \)'s behavior up to 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, \( D_{\theta^i \theta^j} \eta \), and \( D_{\theta^i \theta^j} \eta \). Then CoPG follows the parameter updates in Eq[6] and updates \( (\theta^i, \theta^j) \), 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
We conclude that opponent parameter learning is an effective approach to estimate online the opponent’s policy 
and in Apx. 11 we show how CoPG can be accompanied with all the mentioned variants.

### 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 considers the game theoretic nature of this optimization, and again ignores the players’ interactions. In contrast, LTRPO operates in epochs.

Let

\[
\hat{\theta}^j = \arg\max_{\theta^j} E_{\tau \sim f(\cdot; \theta^1, \theta^2)} \sum_{k=0}^{t-1} \log \pi (\alpha^j_k | s_k, \theta^j)
\]  

Then, the agent utilizes \( \hat{\theta}^j \) to derive its policy updates in Eq. 7 in place of \( \theta^j \). 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, 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,

\[
\eta(\theta^1, \theta^2) = \eta(\theta^1, \theta^2) + E_{\tau \sim f(\cdot; \theta^1, \theta^2)} \sum_{k=0}^{t-1} \gamma^k A(s_k, a_k^1, a_k^2, \theta^1, \theta^2).
\]  

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 \( (\theta^1, \theta^2) \) in the vicinity of \( \theta^1, \theta^2 \). 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, \( D_{KL}(\theta^1, \theta^2) \), \( (\theta^1, \theta^2) \leq \delta \), i.e.,

\[
\max_{\theta^1 \in \Theta^1, \theta^2 \in \Theta^2} \min L_{\theta^1, \theta^2}(\theta^1, \theta^2), \text{with } D_{KL}(\theta^1, \theta^2) \leq \delta.
\]

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. 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 1), 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.

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 \((\mu_1, \sigma_1), (\mu_2, \sigma_2)\) 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. 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

---

1Link to Videos: https://sites.google.com/view/rll-copo

2Traditionally, fictitious and counterfactual regret minimization approaches have been deployed [Neller and Lanctot, 2013].
Markov soccer game, Fig. 2 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 [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 TRC 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 MADDPRG [Lowe et al. [2017] and LOLA [Foerster et al. [2017b]]. We observe that the MADDPRG policy 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].

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 (\(\theta\) player) plays a game with a discriminator \(D(\phi\text{-player})\), i.e.,

\[
\min_{\theta} \max_{\phi} E_{\tau_0}[\log D_\phi(s,a)] + E_{\tau_1}[\log (1 - D_\phi(s,a))] - \lambda H(\theta),
\]

where \(\tau_0\) 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 (\(t^{\text{lap}}\)), and:

\[
\zeta = E_{\tau_0} \left[ \sum_{k=1}^{t^{\text{lap}}} \| a_k - a_k^c \|^2 / |\tau| \right],
\]

(14)
where \( a_k, a^*_k \) 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](https://youtu.be/DtGWZubjcf4)). The CoPO-based agent achieves the lowest \( \zeta \) value and learns a better imitation policy compared to GDA and BC Table 3.

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

<table>
<thead>
<tr>
<th>( \zeta \times 10^{-2} )</th>
<th>PP_{avg}</th>
<th>CoPO-GAIL</th>
<th>GDA-GAIL</th>
<th>PID</th>
<th>BC</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t^{lap}(s) )</td>
<td>13.07</td>
<td>11.82</td>
<td>12.18</td>
<td>14.67</td>
<td>DNF</td>
</tr>
</tbody>
</table>

**Opponent learning:** We next explore the case where one does not directly have access to the opponent’s parameters and they have to be inferred through interaction with the other agent and the environment. We propose to use CoPO-OP (Sec. 3.2), the opponent learning variant of CoPO.

Fig. [7] shows interaction plots of CoPO-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 CoPO-OP agent with the estimated opponent (Fig. [7a]) shows that the CoPO-OP player learns to seize the ball and interact with the opponent (A2, A3). Fig. [18b] shows the interaction plot of CoPO-OP with CoPO, where we observe that CoPO-OP also learns a policy similar to CoPO, which is able to defend, escape and score goals (A3). When directly competing with CoPO, we observe that CoPO-OP can win 46.5% of the games against CoPO.

The experiment details and numerical results on other setups can be found in Apx. [16]. They show that CoPO-OP achieves the performance of CoPO 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].
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.

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

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, Féraot 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, Forster 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, Bušoniu et al., 2010]. In imperfect information games with known rules, e.g., poker [Moravčík et al., 2017], a series of works study algorithmically computing Nash equilibria [Shalev-Swartz 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 [Forster 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 [Forster 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


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


