Proceedings of Machine Learning Research vol 144:1–13, 2021

Robust Reinforcement Learning: A Constrained Game-theoretic Approach

Jing Yu JING@CALTECH.EDU
Computing and Mathematical Sciences
California Institute of Technology

Clement Gehring CLEMENT@GEHRING.IO
Electrical Engineering and Computer Sciences
Massachusetts Institute of Technology

Florian Schäfer SCHAEFER@CALTECH.EDU
Computing and Mathematical Sciences
California Institute of Technology

Animashree Anandkumar ANIMA@CALTECH.EDU
Computing and Mathematical Sciences
California Institute of Technology

Abstract

Reinforcement learning (RL) methods provide state-of-art performance in complex control tasks. However, it has been widely recognized that RL methods often fail to generalize due to unaccounted uncertainties. In this work, we propose a game theoretic framework for robust reinforcement learning that comprises many previous works as special cases. We formulate robust RL as a constrained minimax game between the RL agent and an environmental agent which represents uncertainties such as model parameter variations and adversarial disturbances. To solve the competitive optimization problems arising in our framework, we propose to use competitive mirror descent (CMD). This method accounts for the interactive nature of the game at each iteration while using Bregman divergences to adapt to the global structure of the constraint set. leveraging Lagrangian duality, we demonstrate an RRL policy gradient algorithm based on CMD. We empirically show that our algorithm is stable for large step sizes, resulting in faster convergence on constrained linear quadratic games.

Keywords: robust reinforcement learning, zero-sum game, adversarial training, competitive optimization, policy gradient

1. Introduction

Robustness is crucial for RL. Environmental uncertainties such as unmodeled dynamics, disturbances, and variations of model parameters, are ubiquitous in real-world control tasks (Zhou and Doyle, 1998; Lötjens et al., 2019). It is crucial for reinforcement learning (RL) agents to learn policies robust to environmental uncertainties for them to be reliable (Christiano et al., 2016; Garcıa and Fernández, 2015). For example, policies trained on simulations should account for uncertainties arising from the gap between simulation and reality. Similarly, RL policies trained on data from

a *single* robot/vehicle, might be deployed on a *fleet* of robots, and therefore should account for variations in the manufacturing process. The robustness issue becomes especially important for model-free RL methods that are often trained exclusively on simulations due to their poor sample efficiency. To allow for their safe deployment, it is mandatory for them to be robust to the differences between simulation and real world.

Adversarial training. To improve robustness, robust reinforcement learning (RRL) introduces adversarial attacks such as disturbances during training (Pinto et al., 2017; Tessler et al., 2019). This is generally referred to as adversarial training, where an adversary is modeled against the RL agent to generate adversarial learning conditions. Previous works differ mostly in the modeling of the adversary, the type of policy gradient method employed, and the optimization algorithms.

We provide a unifying perspective on these methods by proposing a general constrained competitive game between the RL agent and an environmental agent that learns to create adversarial environmental uncertainties. In particular, our game-theoretical framework has the flexibility of modeling either a *static* or *dynamic* environmental agent, where a *dynamic* agent's actions depend on current observations while a *static* environmental agent learns a static distribution of some environmental parameters. Moreover, our framework can incorporate general, possibly non-convex constraints on both the RL and environmental agents.

Algorithms for competitive optimization. Classical RL agents can be trained by solving a minimization problem using policy gradient descent. In contrast, our adversarial formulation requires solving a competitive optimization problem where the RL agent tries to minimize its cost, while the environmental agent tries to maximize it. While often used in practice, simultaneous (policy) gradient descent can be shown to diverge even on simple toy problems. To address these issues, numerous methods such as opponent learning awareness (LOLA) (Foerster et al., 2017), optimism (Daskalakis et al., 2017), extragradient methods (Korpelevich, 1977; Gidel et al., 2018), or two-timescale update rules (Heusel et al., 2017) have been proposed.

The iterates of *competitive gradient descent* (CGD) (Schäfer and Anandkumar, 2019) are obtained as Nash equilibria of a local bilinear approximation of the objective function and therefore explicitly account for the interactive nature of the game. Applying CGD to multi-agent reinforcement learning, Prajapat et al. (2020) observe that it learns more sophisticated and efficient policies than simultaneous policy gradient. Schäfer et al. (2020) extend CGD to *competitive mirror descent* (CMD) that uses Bregman divergences to incorporate a wide range of constraints central to control applications including the positive, second order, and positive definite cones. The explicit treatment of player interactions and flexible inclusion of constraints makes CMD a natural tool to solve competitive optimization problems in our framework.

Our contribution. **(1):** We formulate a general framework for RRL as a constrained minimax game that provides a unified perspective to a variety of robust reinforcement learning research. Our framework allows to explicitly promote specific robustness properties in the RL agent and can be combined with most policy gradient algorithms. **(2):** We provide novel gradient and Hessian estimation results (Lemma 1) that are applicable in general RRL settings. **(3):** We develop an RRL algorithm based on competitive mirror descent (CMD) of Schäfer et al. (2020) to solve the constrained minimax game arising from the proposed framework. Independent of the choice of

CMD as the optimization method, we show how to incorporate the Lagrangian formulation to handle general constraints for the agents in our game-theoretical framework.

Related work. In linear control theory, there is well-established connection between differential games and robust control (Başar and Bernhard, 2008). On the other hand, as early as Littman (1994), minimax games were explored for enhancing robustness of RL agents. In (Morimoto and Doya, 2005), an actor-critic network is used to optimize both the RL agent and a dynamic adversary with robust control inspired objective function for the minimax game. Pinto et al. (2017) assume that uncertainties enter the environment as disturbance forces acting on predefined locations and train the RL agent and a dynamic adversary with projected gradient descent ascent. In the same setting, Kamalaruban et al. (2020) use Langevin dynamics for sampling and optimization. In a similar vein, Tessler et al. (2019) treat the adversary as an additive disturbance on the actions taken by the RL agent and solves a minimax game with policy iteration. Deviating from the exact formulation of a minimax game, Mehta et al. (2019) seek to sample difficult uncertainty parameters for the RL agent to learn by requiring an additional reference environment and provide a surrogate objective function for the adversary agent that is optimized using Stein variational policy gradient. Rajeswaran et al. (2016) also sample an ensemble of "worst-case" trajectories defined by conditional value at risk and have the RL agent learn on these trajectories. Both Rajeswaran et al. (2016) and Mehta et al. (2019) consider a static adversary whose policy does not dynamically depend on observations or actions taken by the RL agent. Our approach is also related to distributionally robust optimization (Rahimian and Mehrotra, 2019; Derman and Mannor, 2020) where the minimax problem is not directly solved but rather implicitly incorporated through regularization. Finally, we point out the connection of this work to (Prajapat et al., 2020) where competitive gradient descent (CGD) (Schäfer and Anandkumar, 2019) is applied to multi-agent RL.

2. A constrained minimax game framework for RRL

We consider *uncertain* Markov Decision Processes defined by tuples of the form $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}_\mu, \mathcal{P}_0, \gamma \rangle$ where $\mathcal{S}$, $\mathcal{A}$ are continuous states and actions space respectively, with reward function $\mathcal{R}$ and state transition probability $\mathcal{T}_\mu$ with uncertainties parametrized by μ; initial state distribution $\mathcal{P}_0$, and discount factor $\gamma \in [0, 1]$. An episode starts with an initial state $s_0 \sim \mathcal{P}_0(s_0)$ and at each time step t, transitions to the next state as $s_{t+1} \sim \mathcal{P}_\mu(s_{t+1}|s_t, a_t)$ with uncertainty parameter μ after receiving action a_t. Denote the RL agent's policy as $\pi_\theta(a_t|s_t)$ parametrized by θ. For a fixed time horizon $T + 1$, a trajectory $\tau := \{s_0, a_0, s_1, a_1, \ldots, s_{T+1}\}$ is therefore jointly distributed as $\mathcal{P}_{\mu,\theta}(\tau) := \mathcal{P}_0(s_0)\Pi_{t=0}^{T}\mathcal{P}_\mu(s_{t+1}|s_t, a_t)\pi_\theta(a_t|s_t)$. The RL agent leans a policy $\pi_\theta(a_t|s_t)$ robust against the environmental agent that decides on parameter μ.

We cast RRL as a constrained minimax game between the RL agent's policy parameterized by θ and an environmental agent's policy parameterized by ξ for the environmental uncertainty parameter μ:

$$\min_{\theta \in \Theta} \max_{\xi \in \Xi} \mathbb{E}_{\mu \sim p_\xi(\mu); \tau \sim \mathcal{P}_{\mu,\theta}(\tau)} \left[r(\tau) \right]. \tag{1}$$

where Θ and Ξ are general constraint sets and $p_\xi(\mu)$ is either a static distribution over environmental parameter μ or a dynamic policy. Note that the objective of the RL agent is to *minimize* the *cost* $r(\tau) := \sum_{t=0}^{T} r_t(a_t, s_t)$ whereas the objective of the environmental agent is the opposite. We now discuss three important aspects of formulation (1).

2.1. Modeling of the Environmental Agent

Formulation (1) subsumes the design principles of many RRL methods. In the case of a static environmental agent as in (Mehta et al., 2019; Rajeswaran et al., 2016), the environmental agent's policy is $p_\xi(\mu) := \mathcal{P}_\xi(\mu)$, a static distribution parameterized by ξ over the uncertain environmental parameter μ. Such static environmental agent can represent model parameter uncertainties. In the dynamic case as treated by Pinto et al. (2017); Kamalaruban et al. (2020), the environmental agent explicitly learns a dynamic policy $p_\xi(\mu) := \pi_\xi(\mu_t \,|\, s_t)$ that acts according to the observations. Such dynamic environmental agents can model adversarial disturbances and counteracting forces.

2.2. Policy Gradient Methods

In most RL applications, the game (1) is non-convex and therefore need not admit a canonical solution such as a Nash equilibrium. Nevertheless, we can use it to guide the learning of the RL agent and update the players' policies by descending and ascending along their policy gradients. For most RL tasks, the transition probability and cost function are not accessible. Therefore, we present a gradient estimation result whose derivation is deferred to the Appendix in the full version of the paper online. In the following, we denote the cost function in (1) as $J(\theta, \xi) := \mathbb{E}_{\mu \sim p_\xi(\mu); \tau \sim \mathcal{P}_{\mu,\theta}(\tau)}[r(\tau)]$.

Lemma 1 *When the environmental agent is a static agent $\mu \sim \mathcal{P}_\xi(\mu)$, the gradient and mixed Hessian of $J(\theta, \xi)$ with respect to θ and ξ are:*

$$\nabla_\theta J(\theta,\xi) = \mathbb{E}_{\mu\sim\mathcal{P}_\xi(\mu)}\left[\mathbb{E}_{\tau\sim\mathcal{P}_{\mu,\theta}(\tau)}\left[r(\tau)\sum_{t=0}^{T}\nabla_\theta \log \pi_\theta(a_t \,|\, s_t) \,|\, \mu\right]\right] \tag{2a}$$

$$\nabla_\xi J(\theta,\xi) = \mathbb{E}_{\mu\sim\mathcal{P}_\xi(\mu)}\left[\nabla_\xi \log \mathcal{P}_\xi(\mu)\cdot \mathbb{E}_{\tau\sim\mathcal{P}_{\theta,\mu}(\tau)}\left[r(\tau)\,|\,\mu\right]\right] \tag{2b}$$

$$D^2_{\theta\xi} J(\theta,\xi) = \mathbb{E}_{\mu\sim\mathcal{P}_\xi(\mu)}\left[\nabla_\xi \log \mathcal{P}_\xi(\mu)\cdot \mathbb{E}_{\tau\sim\mathcal{P}_{\mu,\theta}(\tau)}\left[r(\tau)\sum_{t=0}^{T}\nabla_\theta \log \pi_\theta(a_t \,|\, s_t) \,|\, \mu\right]\right]. \tag{2c}$$

When the environmental agent is dynamic with policy $\mu_t \sim \pi_\xi(\mu_t \,|\, s_t)$ where the environmental agent's decision is independent of the current action of the RL agent. Let τ' denote augmented trajectory $\{s_0, a_0, \mu_0, \ldots, s_T\}$ distributed according to

$$\mathcal{P}_{\xi,\theta}(\tau') := \mathcal{P}_0(s_0)\Pi_{t=0}^{T}\mathcal{P}(s_{t+1}|s_t, a_t, \mu_t)\pi_\theta(a_t|s_t)\pi_\xi(\mu_t|s_t),$$

the policy gradients are:

$$\nabla_\theta J(\theta,\xi) = \mathbb{E}_{\tau'\sim\mathcal{P}_{\xi,\theta}(\tau')}\left[r(\tau)\sum_{t=0}^{T}\nabla_\theta \log \pi_\theta(a_t \,|\, s_t)\right] \tag{3a}$$

$$\nabla_\xi J(\theta,\xi) = \mathbb{E}_{\tau'\sim\mathcal{P}_{\xi,\theta}(\tau')}\left[r(\tau)\sum_{t=0}^{T}\nabla_\xi \log \pi_\xi(\mu_t \,|\, s_t)\right] \tag{3b}$$

$$D^2_{\theta\xi} J(\theta,\xi) = \mathbb{E}_{\tau'\sim\mathcal{P}_{\xi,\theta}(\tau')}\left[r(\tau)\sum_{t=0}^{T}\nabla_\theta \log \pi_\theta(a_t \,|\, s_t)\cdot\sum_{t=0}^{T}\nabla_\xi \log \pi_\xi(\mu_t \,|\, s_t)\right]. \tag{3c}$$

where $r(\tau) = \sum_{t=0}^{T} r_t(a_t, s_t)$.

The expectation in Lemma 1 can be approximated by averaging over independently sampled trajectories τ for the present policies. Note that Lemma 1 is an analog of the single-agent policy gradient theorem (Sutton et al., 2000). Therefore, established methods for variance reduction such as the subtraction of a baseline can be readily applied. One can also replace $r(\tau)$ with an estimated Q-function $\hat{Q}^{\pi_\theta, p_\xi(\mu)}(s_t, a_t, \mu_t)$ or variants of advantage functions $\hat{A}^{\pi_\theta, p_\xi(\mu)}(s_t, a_t, \mu_t)$ such as the t-step TD residual and generalized advantage estimation (Schulman et al., 2015). Lemma 1 is applicable to general policy gradient methods under the framework of (1). We point out that similar expressions for the mixed Hessian were derived by Prajapat et al. (2020) in order to apply CGD to RL in two-player games.

2.3. Competitive Optimization Algorithms

Due to the non-convex nature of (1), the choice of competitive optimization algorithm can significantly affect the solution obtained from Equation (1). For example, Schäfer and Anandkumar (2019) provide evidences where bilinear approximation in first-order gradient descent ascent (GDA) significantly improves convergence speed and robustness to the choice of learning rate over simultaneous. Schäfer et al. (2019) show that in GANs training, opponent-aware modeling of the generator and discriminator can significantly stabilize GANs training. Further, Prajapat et al. (2020) observe that policies trained using opponent-aware optimization algorithms are more sophisticated and competitive policies than those trained using GDA variants that myopically optimize each agent's objective.

3. Robust Policies via Competitive Mirror Descent

Under the framework of (1), we develop an RRL algorithm based on competitive mirror descent (CMD) (Schäfer et al., 2020). In what follows, we provide an overview of CMD and present the main algorithm for RRL based on the proposed framework. The introduction of the algorithm is accompanied by an illustrative linear quadratic game example.

3.1. Competitive Mirror Descent

CMD (Schäfer et al., 2020) is a generalization of the mirror descent (Nemirovsky and Yudin, 1983) to the two-player competitive case. Given a constrained minimax problem:

$$\min_{\theta \in \Theta} \max_{\xi \in \Xi} J(\theta, \xi),$$

we define strongly convex and continuously differentiable distance-generating function (DGFs) (Mertikopoulos et al., 2018) for constraint sets Θ and Ξ to be $\Psi_\Theta : \Theta \to \mathbb{R}$ and $\Psi_\Xi : \Xi \to \mathbb{R}$, respectively. Similar to mirror descent, DGFs in CMD are used to inform the local update rule of the global structure of the constraint set, and in particular guaranteeing feasibility of all iterates within the constraint set: $(\theta_k, \xi_k) \in \Theta \times \Xi, \ \forall k$. The CMD updates for $(\theta_{k+1}, \xi_{k+1})$ from the previous iteration (θ_k, ξ_k) at each time step k can be summarized as:

$$\theta_{k+1} = \nabla \Psi_\Theta^{-1} \left(\nabla \Psi_\Theta(\theta_k) + \left[D^2 \Psi_\Theta(\theta_k) \right] \delta_k^\theta \right) \tag{4a}$$

$$\xi_{k+1} = \nabla \Psi_\Theta^{-1} \left(\nabla \Psi_\Xi(\xi_k) + \left[D^2 \Psi_\Xi(\xi_k) \right] \delta_k^\xi \right), \tag{4b}$$

where $\nabla \Psi_\Theta(\theta_k)$ is the DGF function Ψ_Θ evaluated at θ_k and D^2 denotes the Hessian of a function. $\nabla \Psi_\Theta^{-1}(z)$ is the preimage of z under $\nabla \Psi_\Theta$. Since DGFs are strongly convex and continuously

differentiable, the preimage is unique. Moreover, $(\delta_k^\theta, \delta_k^\xi)$ is the Nash equilibrium to a *local, bilinear* approximation of (1) around (θ_k, ξ_k) computed as follows:

$$\delta_k^\theta = \text{argmin}_{\delta_k^\theta} \quad \left\langle \nabla_\theta J(\theta_k, \xi_k),\, \delta_k^\theta \right\rangle + \delta_k^{\theta T} \left[D^2_{\theta\xi} J(\theta_k, \xi_k) \right] \delta_k^\xi + \frac{1}{2\eta_\theta} \delta_k^{\theta T} \left[D^2 \Psi_\Theta(\theta_k) \right] \delta_k^\theta \tag{5}$$

$$\delta_k^\xi = \text{argmax}_{\delta_k^\xi} \quad \left\langle \nabla_\xi J(\theta_k, \xi_k),\, \delta_k^\xi \right\rangle + \delta_k^{\xi T} \left[D^2_{\xi\theta} J(\theta_k, \xi_k) \right] \delta_k^\theta - \frac{1}{2\eta_\xi} \delta_k^{\xi T} \left[D^2 \Psi_\Xi(\xi_k) \right] \delta_k^\xi, \tag{6}$$

where the learning rate parameters (η_θ, η_ξ) regularize the difference between iterates on the constraint manifold specified by the DGFs. The unique Nash equilibrium of (5) has the following closed form:

$$\begin{aligned} \delta_k^\theta &= -\Big(\frac{1}{\eta_\theta} \left[D^2 \Psi_\Theta \right] + \eta_\xi \left[D^2_{\theta\xi} J \right] \left[D^2 \Psi_\Xi \right]^{-1} \left[D^2_{\xi\theta} J \right] \Big)^{-1} \Big(\nabla_\theta J + \eta_\xi \left[D^2_{\theta\xi} J \right] \left[D^2 \Psi_\Xi \right]^{-1} \nabla_\xi J \Big) \\ \delta_k^\xi &= \Big(\frac{1}{\eta_\xi} \left[D^2 \Psi_\Xi \right] + \eta_\theta \left[D^2_{\xi\theta} J \right] \left[D^2 \Psi_\Theta \right]^{-1} \left[D^2_{\theta\xi} J \right] \Big)^{-1} \Big(\nabla_\xi J - \eta_\theta \left[D^2_{\xi\theta} J \right] \left[D^2 \Psi_\Theta \right]^{-1} \nabla_\theta J \Big), \end{aligned} \tag{7}$$

where all gradients and Hessians are evaluated at the current step (θ_k, ξ_k).

We remark that common DGFs include the negative log-determinant function $\Psi_{\mathcal{S}^n_{++}}(X) := -\text{logdet}(X)$ for the positive definite matrix cone $\mathcal{S}^n_{++}$ and translated negative Gibbs-Shannon entropy function $\Psi_{[a,b]^n}(x) := \sum_{i=1}^n (x_i - a)\text{log}(x_i - a) + (b - x_i)\text{log}(b - x_i)$ for box constraints $[a, b]^n$.

3.2. Algorithm for RRL

We propose a learning algorithm in Algorithm 1 for RRL based on CMD. In particular, the algorithm uses CMD iterates to numerically solve (1). The output of the algorithm, after N iterations, is the parameter θ_N for the RL policy $\pi_{\theta_N}(a_t \mid s_t)$, robust against an environmental policy ξ_N.

Algorithm 1: Model-free Robust Policy via Competitive Mirror Descent

Input: Model of the RL agent's policy $\pi_\theta(a_t|s_t)$ parameterized by θ ;
Model of the environmental agent's policy $\pi_\xi(\mu_t|s_t)$ (dynamic) or $\mathcal{P}_\xi(\mu)$ (static) ;
Constraint set Θ and Ξ for policy parameters θ and ξ respectively ;
Associated distance-generating functions Ψ_Θ,Ψ_Ξ.
Initialize parameters $(\theta_0$,$\xi_0)$
for $0 \le k \le N-1$ **do**
  Sample trajectories $\{\tau_i\}_{i=1}^M$;
  Estimate the policy gradients and Hessians $\nabla_\theta J$, $\nabla_\xi J$, $D^2_{\xi\theta} J$, $D^2_{\theta\xi} J$;
  Compute local Nash Equilibrium $(\delta_k^\theta, \delta_k^\xi)$ as in (7) with (θ_k, ξ_k) ;
  $\nabla \Psi_\Theta(\theta_{k+1}) = \nabla \Psi_\Theta(\theta_k) - \left[D^2 \Psi_\Theta(\theta_k) \right] \delta_k^\theta$;
  $\nabla \Psi_\Xi(\xi_{k+1}) = \nabla \Psi_\Xi(\xi_k) + \left[D^2 \Psi_\Xi(\xi_k) \right] \delta_k^\xi$;
end
return (θ_N, ξ_N)

To elaborate on the choice of CMD as the main competitive optimization machinery, we discuss two key perspectives of the proposed framework outlined in Section 2.

Player interaction. For competitive games, Schäfer and Anandkumar (2019) point out that even in the unconstrained case, oscillatory behavior and sensitivity to step sizes are commonly observed in first-order methods. To mitigate this problem, CMD considers a bilinear local approximation in (5) to the original game (1) and update the two players by taking a step in the direction of the Nash equilibrium (7) of the local bilinear game (5). The competitive interaction between the two players is explicitly captured through the *mixed* hessian terms in the mirror gradient step (7). On the other hand, (7) utilizes DGFs corresponding to both Agents' constraint sets to adapt the local update rule to the global structure of the constraints. The incorporation of the bilinear approximation make unconstrained minimax games converge faster and more stably than other first- and second-order methods, even when taking the computational complexity of matrix inversion into account (Schäfer and Anandkumar, 2019). By using iterative methods such as conjugate gradient (Shewchuk et al., 1994) and fast Hessian vector products for computing the update (7), CGD has been applied to generative adversarial networks with millions of degrees of freedom (Schäfer et al., 2019).

Constraint handling. The incorporation of general (non-convex) constraints for the RL agent and the environmental agent in (1) is crucial when there are known safety or physical requirements for the RL policy in control applications. Previous works on RRL commonly use projected gradient descent ascent (PGDA) to jointly take gradient steps that are projected back to the constraint sets for the RL and the environmental agent Pinto et al. (2017); Kamalaruban et al. (2020).

On the other hand, mirror descent Beck and Teboulle (2003) has seen success in constrained optimization problems. It exploits the geometry of the constraint sets via the DGFs corresponding to the constraints. By incorporating the DGFs of both agents' objectives, one obtains local updates that respect both local agent interaction and the global structure of the constraint set (Mertikopoulos et al., 2018; Schäfer et al., 2020).

Recall that mirror descent for $\min_{x\in\mathcal{X}} f(x)$ with an associated DGF Φ has the closed-form update: $\nabla\Phi(x_{t+1}) = \nabla\Phi(x_t) - \eta_t \nabla_x f(x_t)$. Similar mirror descent, CMD makes gradient updates (4) in the "dual" space parameterized by the DGFs. When the constraints include parametric (in)equalities without known DGFs, we consider the Lagrangian of (1) with respect to the (in)equality constraints. We augment the two agents with Lagrange multipliers who are either unconstrained or constrained to $\mathcal{S}^n_{++}$ and transform the general constrained minimax game (1) to one that only has constraints with readily available DGFs. This process is illustrated in Section 3.3.

3.3. Linear Quadratic Games: An Illustration

We demonstrate the proposed RRL algorithm with minimax linear quadratic (LQ) games, a class of well-studied dynamic games with deep connections to the $\mathcal{H}_\infty$ robust control theory (Başar and Bernhard, 2008). LQ games have the discrete-time linear dynamics where one player chsoses action $u_t \in \mathbb{R}^m$ while the other chooses action $w_t \in \mathbb{R}^p$:

$$x_{t+1} = Ax_t + Bu_t + Cw_t, \tag{8}$$

with $x_t \in \mathbb{R}^n$ as the state vector. This formulation is closely related to single-player linear quadratic regulator (LQR) problems (Fazel et al., 2018; Al-Tamimi et al., 2007) where $C = 0$. In LQR, a linear RL agent *minimizes* an infinite-horizon quadratic cost: $\mathbb{E}_{x_0\sim\mathcal{P}}\left[\sum_{t=0}^{\infty}(x_t^T Q x_t + u_t^T R^u u_t)\right]$ where $Q \in \mathbb{R}^{n\times n}$ and $R^u \in \mathbb{R}^{m\times m}$ are positive definite.

The two-player dynamics in (8) adds an environmental agent to the LQR formulation where the environmental agent adversarially chooses disturbance w_t that affect the states x_{t+1} through C. LQ games consider the following minimax objective:

$$\inf_{u_{t,t\geq 0}} \sup_{w_{t,t\geq 0}} \mathbb{E}_{x_0\sim\mathcal{P}} \left[\sum_{t=0}^{\infty}(x_t^T Q x_t + u_t^T R^u u_t - w_t^T R^w w_t)\right]. \tag{9}$$

Zhang et al. (2019) shows that the solution to the zero-sum LQ game (9) subject to (8) corresponds to the solution to a mixed $\mathcal{H}_2/\ \mathcal{H}_\infty$ problem (D'Andrea, 1996). Therefore, the LQ game can be interpreted both as finding a *robust* LQR policy against dynamic disturbances and as an optimal controller for a mixed $\mathcal{H}_2/\ \mathcal{H}_\infty$ problem.

Let the policy for the RL agent (who seeks to *minimize* the cost) and the environmental agent (who seeks to maximize the cost) take the form of $u_t = Kx_t$ and $w_t = Lx_t$ with $K \in \mathbb{R}^{m\times n}$ and $L \in \mathbb{R}^{p\times n}$. We denote the objective in (9) to be $C(K, L)$ to emphasize its dependence on the policy parameters. Under the condition $L \in \Omega := \{L \,|\, Q - L^T R^w L \succ 0\}$ and other technical assumptions (Zhang et al., 2019), a globally unique and obtainable Nash equilibrium of the LQ game exists. We can pose an equivalent constrained minimax game for (9) as:

$$\min_K \max_{L\in\Omega} C(K, L). \tag{10}$$

In this game, the RL agent finds the best linear policy that is robust against the worst dynamic environmental disturbances. The solution to the LQ game can be obtained via linear matrix inequality (D'Andrea, 1996). However, when the system matrices and the objective function are unknown, our proposed framework for robust policy via CMD can be applied.

We first illustrate how to handle conic constraints that are closely related to stability and safety of the resulting policy, such as $L \in \Omega$, in Algorithm 1. Consider the Lagrangian of (10) with Lagrangian multiplier $\Lambda \in \mathcal{S}^n_{++}$:

$$\mathcal{L}(\Lambda, K, L) = C(K, L) - \left\langle \Lambda, L^T R^w L - Q \right\rangle. \tag{11}$$

where $\langle X, Y\rangle := \text{tr}(X^T Y)$ denotes the matrix inner product. We augment the RL agent (K) with the Lagrangian multiplier Λthat penalizes the adversary (L) if it does not satisfy the constraint $L \in \Omega$. Using the Lagrangian as the augmented objective function, we arrive at the following constrained minimax game, conforming to the general formulation proposed in (1), for the proposed RRL algorithm:

$$\min_{K,\Lambda\in\mathcal{S}^n_{++}} \max_L \ \mathcal{L}(\Lambda, K, L). \tag{12}$$

Note that the Lagrangian multiplier that enforces the constraint on the *maximizing* player is assigned to the *minimizing* player. It is straight forward to verify that the solution to (10) is the solution to (12) and vice versa because of the complimentary slackness at KKT points. Intuitively, (12) means that whenever the constraints on the *maximizing* player is not satisfied, the *minimizing* player can improve its objective by increasing the Lagrange multiplier.

We choose the log-determinant function as the associated DGF for the Lagrange multiplier $\Lambda \in \mathcal{S}^n_{++}$. For all other variables in (12) that are unconstrained, matrix Euclidean norm whose derivative is

the identity mapping and the hessian is the identity matrix can be used as the DGF. Following the notations in Algorithm 1, the DGFs for both players are

$$\Psi_\Theta(K,\Lambda) = -\text{logdet}(\Lambda) + \frac{1}{2}\|K\|_F^2 \quad , \quad \Psi_\Xi(L) = \frac{1}{2}\|L\|_F^2.$$

Note that due to the structure of the Lagrangian function $\mathcal{L}$, its partial derivatives and mixed hessians required for Algorithm 1, are decoupled into the *model-free* and *modeled* portions during computation. More specifically, The gradients for the RL agent as the minimizing player and the environmental agent as the maximizing player are

$$\nabla_{(K,\Lambda)}\mathcal{L}(\Lambda,K,L) = \begin{bmatrix} \nabla_K C(K,L) \\ \nabla_\Lambda \langle \Lambda, L^T R^w L - Q \rangle \end{bmatrix}$$
$$\nabla_L \mathcal{L}(\Lambda,K,L) = \nabla_L C(K,L) + \nabla_L \langle \Lambda, L^T R^w L - Q \rangle,$$

where $\nabla_K C(K,L)$ and $\nabla_L C(K,L)$ can be estimated by sampling trajectories per Lemma 1 and $\nabla_\Lambda \langle \Lambda, L^T R^w L - Q \rangle$ and $\nabla_L \langle \Lambda, L^T R^w L - Q \rangle$ are the derivatives of a known constraint. Similar decoupling happens when one takes the mixed Hessians. This feature of separability is instrumental in using Algorithm 1 to solve (1) with arbitrary and potentially non-convex constraints. In general, the constraint sets for the RL agent and the environmental variables are known a priori. This mean the Lagrangian multiplier component of the augmented reward function is known and its gradients can be analytically computed for each agents, independent of the unknown original reward function for the RL agent. Therefore, separability property demonstrated here holds for general constraints and robust RL formulation that falls under the proposed framework

4. Simulation

We now consider a double integrator LQ game with $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, and $C = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$. The quadratic cost has $Q = I$, $R^u = 1$, $R^w = 20$. We randomly generate a stabilizing K_0 and initialize the environmental agent's parameter L_0 in the interior of the constraint set. We sample trajectories of length $T = 15$ as the finite-horizon approximation to the the infinite-horizon LQ cost and compute the corresponding gradient and Hessian estimations.

We compare our method against projected nested gradient descent (PNGD) proposed in Zhang et al. (2019) and projected gradient descent ascent (PGDA) as baseline. The result is shown in Figure 1. For each of the method, we test a variety of steps sizes for both minimizing player and maximizing player varying from 10^{-3} to 10^{-5}. We observe that for steps sizes larger than 10^{-5}, both PNGD with 50 inner loop iteration and PGDA diverges. On the other hand, Figure 2 shows that our method is stable for step sizes larger than the ones tolerable for other presented methods. As in similar experiments by Schäfer and Anandkumar (2019), the bilinear approximation employed in our method facilitates the learning process and results in faster convergence.

Acknowledgments

We thank the anonymous referees for their valuable feedback. CG gratefully acknowledges support from NSF grant 1723381; from AFOSR grant FA9550-17-1-0165; from ONR grant N00014-18-1-2847 and from the MIT-IBM Watson Lab. FS gratefully acknowledges support by the Air Force

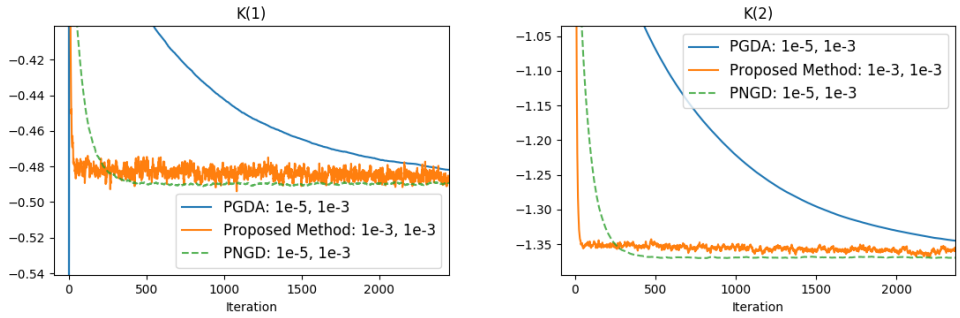

Figure 1: Comparison of Various Methods: We tested step sizes varying from 10^{-3} to 10^{-5} for the proposed algorithm, PNGD with inner loop iteration number set to 10, and PGDA. For each method, we plot the fastest converging trajectory against the number of *outer* iterations. The two step sizes are specified for minimizing player and maximizing player, respectively. Optimal closed-form solution is $K^* = [-0.4913, -1.3599]^T$.

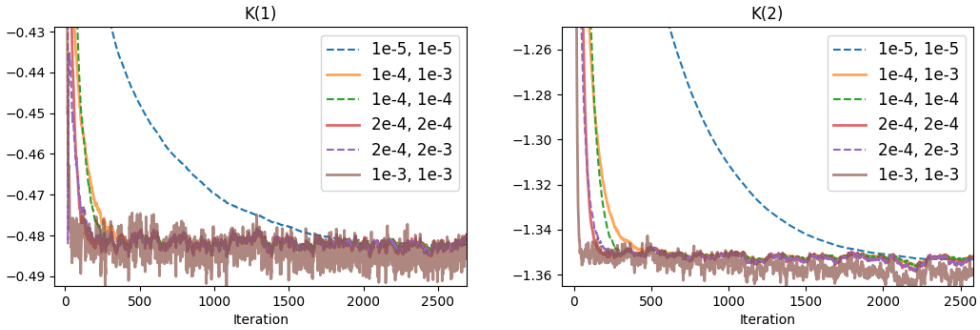

Figure 2: Various Step Sizes for the Proposed Algorithm: The proposed algorithm based on CMD is robust to a large range of step sizes. The two panels show the iteration trajectory for the coordinates of parameter K. The two step sizes are specified for minimizing player and maximizing player, respectively. Optimal closed-form solution is $K^* = [-0.4913, -1.3599]^T$.

Office of Scientific Research under award number FA9550-18-1-0271 (Games for Computation and Learning) and the Ronald and Maxine Linde Institute of Economic and Management Sciences at Caltech. AA is supported in part by the Bren endowed chair, Microsoft, Google, Facebook and Adobe faculty fellowships.

5. Conclusion and Outlook

We propose a constrained minimax game between the reinforcement learning agent and a modelled environmental agent as a unifying perspective on robust reinforcement learning. We develop a learning algorithm based on competitive mirror descent to solve the minimax game arises from our framework. The algorithm employs Lagrangian duality and Bregman divergences to handle general constraints and inherits robustness against large step sizes from competitive mirror descent. We demonstrate this feature on numerical experiments on linear quadratic games. In future work we plan to study the performance of our approach on high-dimensional control tasks.

References

Asma Al-Tamimi, Frank L Lewis, and Murad Abu-Khalaf. Model-free q-learning designs for linear discrete-time zero-sum games with application to h-infinity control. *Automatica*, 43(3):473–481, 2007.

Tamer Başar and Pierre Bernhard. *H-infinity optimal control and related minimax design problems: a dynamic game approach*. Springer Science & Business Media, 2008.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Paul Christiano, Zain Shah, Igor Mordatch, Jonas Schneider, Trevor Blackwell, Joshua Tobin, Pieter Abbeel, and Wojciech Zaremba. Transfer from simulation to real world through learning deep inverse dynamics model. *arXiv preprint arXiv:1610.03518*, 2016.

Raffaello D'Andrea. Lmi approach to mixed h-2 and h-infinity performance objective controller design. *IFAC Proceedings Volumes*, 29(1):3198–3203, 1996.

Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training gans with optimism. *arXiv preprint arXiv:1711.00141*, 2017.

Esther Derman and Shie Mannor. Distributional robustness and regularization in reinforcement learning. *arXiv preprint arXiv:2003.02894*, 2020.

Maryam Fazel, Rong Ge, Sham M Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. *arXiv preprint arXiv:1801.05039*, 2018.

Jakob N Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326*, 2017.

Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Gauthier Gidel, Hugo Berard, Gaëtan Vignoud, Pascal Vincent, and Simon Lacoste-Julien. A variational inequality perspective on generative adversarial networks. *arXiv preprint arXiv:1802.10551*, 2018.

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 *Advances in neural information processing systems*, pages 6626–6637, 2017.

Parameswaran Kamalaruban, Yu-Ting Huang, Ya-Ping Hsieh, Paul Rolland, Cheng Shi, and Volkan Cevher. Robust reinforcement learning via adversarial training with langevin dynamics. *arXiv preprint arXiv:2002.06063*, 2020.

GM Korpelevich. Extragradient method for finding saddle points and other problems. *Matekon*, 13 (4):35–49, 1977.

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

Björn Lötjens, Michael Everett, and Jonathan P How. Safe reinforcement learning with model uncertainty estimates. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8662–8668. IEEE, 2019.

Bhairav Mehta, Manfred Diaz, Florian Golemo, Christopher J Pal, and Liam Paull. Active domain randomization. *arXiv preprint arXiv:1904.04762*, 2019.

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. *arXiv preprint arXiv:1807.02629*, 2018.

Jun Morimoto and Kenji Doya. Robust reinforcement learning. *Neural computation*, 17(2):335–359, 2005.

Arkadiu Semenovich Nemirovsky and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2817–2826. JMLR. org, 2017.

Manish Prajapat, Kamyar Azizzadenesheli, Alexander Liniger, Yisong Yue, and Anima Anandkumar. Competitive policy optimization. *arXiv preprint arXiv:2006.10611*, 2020.

Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. Epopt: Learning robust neural network policies using model ensembles. *arXiv preprint arXiv:1610.01283*, 2016.

Florian Schäfer and Anima Anandkumar. Competitive gradient descent. In *Advances in Neural Information Processing Systems*, pages 7623–7633, 2019.

Florian Schäfer, Hongkai Zheng, and Anima Anandkumar. Implicit competitive regularization in gans. *arXiv preprint arXiv:1910.05852*, 2019.

Florian Schäfer, Anima Anandkumar, and Houman Owhadi. Competitive mirror descent, 2020.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

Jonathan Richard Shewchuk et al. An introduction to the conjugate gradient method without the agonizing pain, 1994.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. *arXiv preprint arXiv:1901.09184*, 2019.

Kaiqing Zhang, Zhuoran Yang, and Tamer Basar. Policy optimization provably converges to nash equilibria in zero-sum linear quadratic games. In *Advances in Neural Information Processing Systems*, pages 11598–11610, 2019.

Kemin Zhou and John Comstock Doyle. *Essentials of robust control*, volume 104. Prentice hall Upper Saddle River, NJ, 1998.