

---

# A Reduction from Reinforcement Learning to No-Regret Online Learning

---

Ching-An Cheng  
Georgia Tech

Remi Tachet des Combes  
Microsoft Research

Byron Boots  
University of Washington

Geoff Gordon  
Microsoft Research

## Abstract

We present a reduction from reinforcement learning (RL) to no-regret online learning based on the saddle-point formulation of RL, by which *any* online algorithm with sublinear regret can generate policies with provable performance guarantees. This new perspective decouples the RL problem into two parts: regret minimization and function approximation. The first part admits a standard online-learning analysis, and the second part can be quantified independently of the learning algorithm. Therefore, the proposed reduction can be used as a tool to systematically design new RL algorithms. We demonstrate this idea by devising a simple RL algorithm based on mirror descent and the generative-model oracle. For any  $\gamma$ -discounted tabular RL problem, with probability at least  $1 - \delta$ , it learns an  $\epsilon$ -optimal policy using at most  $\tilde{O}\left(\frac{|\mathcal{S}||\mathcal{A}|\log(\frac{1}{\delta})}{(1-\gamma)^4\epsilon^2}\right)$  samples. Furthermore, this algorithm admits a direct extension to linearly parameterized function approximators for large-scale applications, with computation and sample complexities independent of  $|\mathcal{S}|, |\mathcal{A}|$ , though at the cost of potential approximation bias.

## 1 INTRODUCTION

Reinforcement learning (RL) is a fundamental problem for sequential decision making in unknown environments. One of its core difficulties, however, is the need for algorithms to infer long-term consequences based on limited, noisy, short-term feedback. As a result, designing RL algorithms that are both scalable and provably sample efficient has been challenging.

In this work, we revisit the classic linear-program (LP) formulation of RL (Manne et al., 1959; Denardo and Fox, 1968) in an attempt to tackle this long-standing question. We focus on the associated saddle-point problem of the LP (given by Lagrange duality), which has recently gained traction due to its potential for computationally efficient algorithms with theoretical guarantees (Wang and Chen, 2016; Chen and Wang, 2016; Wang, 2017a; Lee and He, 2018; Wang, 2017b; Lin et al., 2017; Dai et al., 2018; Chen et al., 2018; Lakshminarayanan et al., 2018). But in contrast to these previous works based on stochastic approximation, here we consider a reformulation through the lens of online learning, i.e. regret minimization. Since the pioneering work of Kivinen and Warmuth (1997); Gordon (1999); Zinkevich (2003), online learning has evolved into a ubiquitous tool for systematic design and analysis of iterative algorithms. Therefore, if we can identify a reduction from RL to online learning, we can potentially leverage it to build efficient RL algorithms.

We will show this idea is indeed feasible. We present a reduction by which *any* no-regret online algorithm, after observing  $N$  samples, can find a policy  $\hat{\pi}_N$  in a policy class  $\Pi$  satisfying  $V^{\hat{\pi}_N}(p) \geq V^{\pi^*}(p) - o(1) - \epsilon_\Pi$ , where  $V^\pi(p)$  is the accumulated reward of policy  $\pi$  with respect to some unknown initial state distribution  $p$ ,  $\pi^*$  is the optimal policy, and  $\epsilon_\Pi \geq 0$  is a measure of the expressivity of  $\Pi$  (see Section 4.2 for definition).

Our reduction is built on a refinement of online learning, called Continuous Online Learning (COL) (Cheng et al., 2019a), which was proposed to model problems where loss gradients across rounds change continuously with the learner’s decisions. COL has a strong connection to equilibrium problems (EPs) (Blum, 1994; Bianchi and Schaible, 1996), and any monotone EP (including our saddle-point problem of interest) can be framed as no-regret learning in a properly constructed COL problem (Cheng et al., 2019a). Using this idea, our reduction follows naturally by first converting an RL problem to an EP and then the EP to a COL problem.

Framing RL as COL reveals new insights into the relationship between approximate solutions to the saddle-

point problem and approximately optimal policies. Importantly, this new perspective shows that the RL problem can be separated into two parts: regret minimization and function approximation. The first part admits standard treatments from the online learning literature, and the second part can be quantified *independently* of the learning process. For example, one can accelerate learning by adopting optimistic online algorithms (Rakhlin and Sridharan, 2012; Cheng et al., 2019b) that account for the predictability in COL, without worrying about function approximators. Because of these problem-agnostic features, the proposed reduction can be used to systematically design efficient RL algorithms with performance guarantees.

As a demonstration, we design an RL algorithm based on a simple online learning algorithm: mirror descent. Assuming a generative model<sup>1</sup> we first prove that, for *any* tabular Markov decision process (MDP), with probability at least  $1 - \delta$ , this algorithm learns an  $\epsilon$ -optimal policy for the  $\gamma$ -discounted accumulated reward, using at most  $\tilde{O}\left(\frac{|\mathcal{S}||\mathcal{A}|\log(\frac{1}{\delta})}{(1-\gamma)^4\epsilon^2}\right)$  samples, where  $|\mathcal{S}|, |\mathcal{A}|$  are the sizes of state and action spaces, and  $\gamma$  is the discount rate. Next, we use the separation property above and present a natural extension based on linearly parameterized function approximators, which is also applicable to continuous problems. This version has sample and per-round computational complexities linear in the number of parameters, *independent* of  $|\mathcal{S}|, |\mathcal{A}|$ , though at the cost of policy performance bias due to approximation.

This new sample complexity improves the current best provable rate of the saddle-point RL setup (Wang and Chen, 2016; Chen and Wang, 2016; Wang, 2017a; Lee and He, 2018) by a large factor of  $\frac{|\mathcal{S}|^2}{(1-\gamma)^2}$ , *without* making any assumption on the MDP.<sup>2</sup> This improvement is attributed to our new online-learning-style analysis that uses a cleverly selected comparator in the regret definition. While it is possible to devise a minor modification of the previous stochastic mirror descent algorithms, e.g. Wang (2017a), achieving the same rate with our new analysis, we remark that our algorithm is considerably simpler and removes a projection required in previous work (Wang and Chen, 2016; Chen and Wang, 2016; Wang, 2017a; Lee and He, 2018).

Finally, we do note that the same sample complexity can also be achieved, e.g., by model-based RL and (phased) Q-learning (Kearns and Singh, 1999; Kakade et al., 2003). However, these methods either have super-linear runtime, or could become unstable when using

<sup>1</sup>In practice, a generative model can be approximated by running a behavior policy with sufficient exploration (Kearns and Singh, 1999).

<sup>2</sup>Wang (2017a) has the same sample complexity but requires the MDP to be ergodic under any policy.

function approximators without further assumptions.

## 2 SETUP & PRELIMINARIES

Let  $\mathcal{S}$  and  $\mathcal{A}$  be state and action spaces, which can be discrete or continuous. We consider  $\gamma$ -discounted infinite-horizon problems for  $\gamma \in [0, 1)$ . Our goal is to find a policy  $\pi(a|s)$  that maximizes the discounted average return  $V^\pi(p) := \mathbb{E}_{s \sim p}[V^\pi(s)]$ , where

$$V^\pi(s) := (1 - \gamma)\mathbb{E}_{\xi \sim \rho^\pi(s)}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)] \quad (1)$$

is the value function of  $\pi$  at state  $s$ ,  $p$  the initial state distribution,  $r : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$  the reward function, and  $\rho^\pi(s)$  the distribution of trajectory  $\xi = s_0, a_0, s_1, \dots$  generated by running  $\pi$  from  $s_0 = s$  in an MDP. We assume that the initial distribution  $p(s_0)$ , the transition  $\mathcal{P}(s'|s, a)$ , and the reward function  $r(s, a)$  are unknown but can be queried through a *generative model*, i.e. we can sample  $s_0$  from  $p$ ,  $s'$  from  $\mathcal{P}$ , and  $r(s, a)$  for any  $s \in \mathcal{S}$  and  $a \in \mathcal{A}$ . We remark that the definition of  $V^\pi$  in (1) contains a  $(1 - \gamma)$  factor. We adopt this setup to make writing more compact. We denote the optimal policy as  $\pi^*$  and its value function as  $V^*$  for short.

### 2.1 Duality in RL

Our reduction is based on the linear-program (LP) formulation of RL. We provide a short recap here (please see Appendix A and Puterman (2014) for details). To show how  $\max_\pi V^\pi(p)$  can be framed as a LP, let us define the average state distribution under  $\pi$ ,  $d^\pi(s) := (1 - \gamma)\sum_{t=0}^{\infty} \gamma^t d_t^\pi(s)$ , where  $d_t^\pi$  is the state distribution at time  $t$  generated by running  $\pi$  from  $p$ . By construction,  $d^\pi$  satisfies the stationarity property,

$$d^\pi(s') = (1 - \gamma)p(s') + \gamma\mathbb{E}_{s \sim d^\pi}\mathbb{E}_{a \sim \pi|s}[\mathcal{P}(s'|s, a)]. \quad (2)$$

With  $d^\pi$ , we can write  $V^\pi(p) = \mathbb{E}_{s \sim d^\pi}\mathbb{E}_{a \sim \pi|s}[r(s, a)]$  and our objective  $\max_\pi V^\pi(p)$  equivalently as:

$$\begin{aligned} \max_{\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}; \boldsymbol{\mu} \geq \mathbf{0}} \quad & \mathbf{r}^\top \boldsymbol{\mu} \\ \text{s.t.} \quad & (1 - \gamma)\mathbf{p} + \gamma\mathbf{P}^\top \boldsymbol{\mu} = \mathbf{E}^\top \boldsymbol{\mu} \end{aligned} \quad (3)$$

where  $\mathbf{r} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ ,  $\mathbf{p} \in \mathbb{R}^{|\mathcal{S}|}$ , and  $\mathbf{P} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|}$  are vector forms of  $r$ ,  $p$ , and  $\mathcal{P}$ , respectively, and  $\mathbf{E} = \mathbf{I} \otimes \mathbf{1} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|}$  (we use  $|\cdot|$  to denote the cardinality of a set,  $\otimes$  the Kronecker product,  $\mathbf{I} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$  is the identity, and  $\mathbf{1} \in \mathbb{R}^{|\mathcal{A}|}$  the vector of ones). In (3),  $\mathcal{S}$  and  $\mathcal{A}$  may seem to have finite cardinalities, but the same formulation extends to countable continuous spaces under proper regularity assumptions (Hernández-Lerma and Lasserre, 2012). We adopt this abuse of notation for compactness.

The variable  $\boldsymbol{\mu}$  of the LP in (3) resembles the joint distribution  $d^\pi(s)\pi(a|s)$ . To see this, notice that the constraint in (3) is reminiscent of (2), and implies  $\|\boldsymbol{\mu}\|_1 = 1$ , i.e.  $\boldsymbol{\mu}$  is a probability distribution.

Then one can show  $\mu(s, a) = d^\pi(s)\pi(a|s)$  when the constraint is satisfied, which implies that (3) is the same as  $\max_\pi V^\pi(p)$  and its solution  $\boldsymbol{\mu}^*$  corresponds to  $\mu^*(s, a) = d^{\pi^*}(s)\pi^*(a|s)$  of the optimal policy  $\pi^*$ .

As (3) is a LP, it suggests looking at its dual, which turns out to be the classic LP formulation of RL<sup>3</sup>,

$$\begin{aligned} \min_{\mathbf{v} \in \mathbb{R}^{|S|}} \quad & \mathbf{p}^\top \mathbf{v} \\ \text{s.t.} \quad & (1 - \gamma)\mathbf{r} + \gamma \mathbf{P}\mathbf{v} \leq \mathbf{E}\mathbf{v}. \end{aligned} \quad (4)$$

It can be verified that for all  $\mathbf{p} > 0$ , the solution to (4) satisfies the Bellman equation (Bellman, 1954) and therefore is the optimal value function  $\mathbf{v}^*$  (the vector form of  $V^*$ ). We note that, for any policy  $\pi$ ,  $V^\pi$  by definition satisfies a stationarity property

$$V^\pi(s) = \mathbb{E}_{a \sim \pi|s} [(1 - \gamma)r(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}|s, a} [V^\pi(s')]] \quad (5)$$

which can be viewed as a dual equivalent of (2) for  $d^\pi$ . Because, for any  $s \in \mathcal{S}$  and  $a \in \mathcal{A}$ ,  $r(s, a)$  is in  $[0, 1]$ , (5) implies  $V^\pi(s)$  lies in  $[0, 1]$  too.

## 2.2 Toward RL: the Saddle-Point Setup

The LP formulations above require knowing the probabilities  $p$  and  $\mathcal{P}$  and are computationally inefficient. If only generative models are available (as in our setup), one can alternatively exploit the duality between the two LPs in (3) and (4), and frame RL as a saddle-point problem (Wang and Chen, 2016). Let us define

$$\mathbf{a}_\mathbf{v} := \mathbf{r} + \frac{1}{1-\gamma}(\gamma \mathbf{P} - \mathbf{E})\mathbf{v} \quad (6)$$

as the *advantage function* with respect to  $\mathbf{v}$  ( $\mathbf{v}$  is not necessarily a value function). Then the saddle-point formulation of the two LPs, using the associated Lagrangian  $\mathcal{L}$ , can be written as

$$\min_{\mathbf{v} \in \mathcal{V}} \max_{\boldsymbol{\mu} \in \mathcal{M}} \mathcal{L}(\mathbf{v}, \boldsymbol{\mu}), \quad (7)$$

where  $\mathcal{L}(\mathbf{v}, \boldsymbol{\mu}) := \mathbf{p}^\top \mathbf{v} + \boldsymbol{\mu}^\top \mathbf{a}_\mathbf{v}$  and the constraints are

$$\mathcal{V} = \{\mathbf{v} \in \mathbb{R}^{|S|} : \mathbf{v} \geq 0, \|\mathbf{v}\|_\infty \leq 1\} \quad (8)$$

$$\mathcal{M} = \{\boldsymbol{\mu} \in \mathbb{R}^{|S||\mathcal{A}|} : \boldsymbol{\mu} \geq 0, \|\boldsymbol{\mu}\|_1 = 1\}. \quad (9)$$

The solution to (7) is exactly  $(\mathbf{v}^*, \boldsymbol{\mu}^*)$ . Extra constraints on the norm of  $\boldsymbol{\mu}$  and  $\mathbf{v}$  are introduced in  $\mathcal{V}, \mathcal{M}$  (compared to (3) and (4)). This common practice uses known bounds on the solutions of (3) and (4) to make the search spaces  $\mathcal{V}$  and  $\mathcal{M}$  compact and small, so that optimization converges faster.

Having compact variable sets allows using first-order stochastic methods, such as stochastic mirror descent and mirror-prox (Nemirovski et al., 2009; Juditsky et al., 2011), to efficiently solve the problem. These methods only require using the generative model to

<sup>3</sup>Our setup in (4) differs from the classic one in the  $(1 - \gamma)$  factor in the constraint due to the average setup.

compute unbiased estimates of the gradients  $\nabla_{\mathbf{v}}\mathcal{L} = \mathbf{b}_\boldsymbol{\mu}$  and  $\nabla_{\boldsymbol{\mu}}\mathcal{L} = \mathbf{a}_\mathbf{v}$ , where we define

$$\mathbf{b}_\boldsymbol{\mu} := \mathbf{p} + \frac{1}{1-\gamma}(\gamma \mathbf{P} - \mathbf{E})^\top \boldsymbol{\mu} \quad (10)$$

as the *balance function* with respect to  $\boldsymbol{\mu}$ .  $\mathbf{b}_\boldsymbol{\mu}$  measures whether  $\boldsymbol{\mu}$  violates the stationarity constraint in (3) and can be viewed as the dual of  $\mathbf{a}_\mathbf{v}$ . When the state or action space is too large, one can resort to function approximators to represent  $\mathbf{v}$  and  $\boldsymbol{\mu}$  (Chen et al., 2018).

## 2.3 COL and EPs

Finally, we review the COL setup in (Cheng et al., 2019a), which we will use to design the reduction from the saddle-point problem in (7) to online learning. An online learning problem describes the iterative interactions between a learner and an opponent. In round  $n$ , the learner chooses a decision  $x_n$  from a decision set  $\mathcal{X}$ , the opponent chooses a per-round loss function  $l_n : \mathcal{X} \rightarrow \mathbb{R}$  based on the learner's decisions, and then information about  $l_n$  (e.g. its gradient  $\nabla l_n(x_n)$ ) is revealed to the learner. The performance of the learner is usually measured in terms of regret with respect to some  $x' \in \mathcal{X}$ ,

$$\text{Regret}_N(x') := \sum_{n=1}^N l_n(x_n) - \sum_{n=1}^N l_n(x').$$

When  $l_n$  is convex and  $\mathcal{X}$  is compact and convex, many no-regret (i.e.  $\text{Regret}_N(x') = o(N)$ ) algorithms are available, such as mirror descent and follow-the-regularized-leader (Cesa-Bianchi and Lugosi, 2006; Shalev-Shwartz et al., 2012; Hazan et al., 2016).

COL is a subclass of online learning problems where the loss sequence changes continuously with respect to the played decisions of the learner (Cheng et al., 2019a). In COL, the opponent is equipped with a bifunction  $f : (x, x') \mapsto f_x(x')$ , where for any fixed  $x' \in \mathcal{X}$ ,  $\nabla f_x(x')$  is continuous in  $x \in \mathcal{X}$ . The opponent selects per-round losses based on  $f$ , but the learner does not know  $f$ : in round  $n$ , if the learner chooses  $x_n$ , the opponent sets

$$l_n(x) = f_{x_n}(x), \quad (11)$$

and returns, e.g., a stochastic estimate of  $\nabla l_n(x_n)$  (the regret is still measured in terms of the noise-free  $l_n$ ).

In Cheng et al. (2019a), a natural connection is shown between COL and equilibrium problems (EPs). As EPs include the saddle-point problem of interest, we can use this idea to turn (7) into a COL problem. Recall an EP is defined as follows: Let  $\mathcal{X}$  be compact and  $F : (x, x') \mapsto F(x, x')$  be a bifunction s.t.  $\forall x, x' \in \mathcal{X}$ ,  $F(\cdot, x')$  is continuous,  $F(x, \cdot)$  is convex, and  $F(x, x) \geq 0$ .<sup>4</sup> The problem  $\text{EP}(\mathcal{X}, F)$  aims to find  $x^* \in \mathcal{X}$  s.t.

$$F(x^*, x) \geq 0, \quad \forall x \in \mathcal{X}. \quad (12)$$

<sup>4</sup>We restrict ourselves to this convex and continuous case as it is sufficient for our problem setup.

By its definition, a natural residual function to quantify the quality of an approximation solution  $x$  to EP is  $r_{ep}(x) := -\min_{x' \in \mathcal{X}} F(x, x')$  which describes the degree to which (12) is violated at  $x$ . We say a bifunction  $F$  is *monotone* if,  $\forall x, x' \in \mathcal{X}$ ,  $F(x, x') + F(x', x) \leq 0$ , and *skew-symmetric* if the equality holds.

EPs with monotone bifunctions represent general convex problems, including saddle-point problems, variational inequalities, etc. For instance, a convex-concave problem  $\min_{y \in \mathcal{Y}} \max_{z \in \mathcal{Z}} \phi(y, z)$  can be cast as EP( $\mathcal{X}, F$ ) with  $\mathcal{X} = \mathcal{Y} \times \mathcal{Z}$  and the skew-symmetric bifunction (Jofré and Wets, 2014)

$$F(x, x') := \phi(y', z) - \phi(y, z'), \quad (13)$$

where  $x = (y, z)$  and  $x' = (y', z')$ . In this case,  $r_{ep}(x) = \max_{z' \in \mathcal{Z}} \phi(y, z') - \min_{y' \in \mathcal{Y}} \phi(y', z)$  is the duality gap.

As shown in Cheng et al. (2019a), a learner achieves sublinear dynamic regret in COL if and only if the same algorithm can solve EP( $\mathcal{X}, F$ ) with  $F(x, x') = f_x(x') - f_x(x)$ . Concretely, given a monotone EP( $\mathcal{X}, F$ ) with  $F(x, x) = 0$  (which is satisfied by (13)), one can construct a COL problem by setting  $f_{x'}(x) := F(x', x)$ , i.e.  $l_n(x) = F(x_n, x)$ , such that any no-regret algorithm can generate an approximate solution to the EP.

**Proposition 1.** (Cheng et al., 2019a) *If  $F$  is skew-symmetric and  $l_n(x) = F(x_n, x)$ , then  $r_{ep}(\hat{x}_N) \leq \frac{1}{N} \text{Regret}_N$ , where  $\text{Regret}_N = \max_{x \in \mathcal{X}} \text{Regret}_N(x)$ , and  $\hat{x}_N = \frac{1}{N} \sum_{n=1}^N x_n$ ; the same guarantee holds also for the best decision in  $\{x_n\}_{n=1}^N$ .*

### 3 AN ONLINE LEARNING VIEW

We present an alternate online-learning perspective on the saddle-point formulation in (7). This analysis paves a way for our reduction in the next section. By reduction, we mean realizing the two steps below:

1. Define a sequence of online losses such that any algorithm with sublinear regret can produce an approximate solution to the saddle-point problem.
2. Convert the approximate solution in the first step to an approximately optimal policy in RL.

Methods to achieve these two steps individually are not new. The reduction from convex-concave problems to two-player no-regret online learning is well known (Abernethy et al., 2011). Likewise, the relationship between the approximate solution of (7) and policy performance is also available; this is how the saddle-point formulation (Wang, 2017a) works in the first place. So couldn't we just use these existing approaches? We argue that purely combining these two techniques fails to fully capture important structure that resides in RL. While this will be made precise in the later analyses, we highlight the main insights here.

Instead of treating (7) as an adversarial two-player online learning problem (Abernethy et al., 2011), we adopt the recent reduction to COL (Cheng et al., 2019a) reviewed in Section 2.3. The main difference is that the COL approach takes a single-player setup and retains the Lipschitz continuity in the source saddle-point problem. This single-player perspective provides a simple setup to analyze effects of function approximators, as we will show in Section 4.2. Additionally, due to continuity, the losses in COL are predictable and therefore make designing fast algorithms possible.

With the help of the COL reformulation, we study the relationship between the approximate solution to (7) and the performance of the associated policy in RL. We are able to establish a tight bound between the residual and the performance gap, resulting in a large improvement of  $\frac{|S|^2}{(1-\gamma)^2}$  in sample complexity compared with the best bounds in the literature of the saddle-point setup, *without* adding extra constraints on  $\mathcal{X}$  or assumptions on the MDP. Overall, this means that *stronger* sample complexity guarantees can be attained by *simpler* algorithms, as we demonstrate in Section 5. The missing proofs of this section are in Appendix B.

#### 3.1 The COL Formulation of RL

First, let us exercise the above COL idea with the saddle-point formulation of RL in (7). To construct the EP, we can let  $\mathcal{X} = \{x = (\mathbf{v}, \boldsymbol{\mu}) : \mathbf{v} \in \mathcal{V}, \boldsymbol{\mu} \in \mathcal{M}\}$ , which is compact and convex. By (13), the bifunction  $F$  of the associated EP( $\mathcal{X}, F$ ) is naturally given as

$$\begin{aligned} F(x, x') &:= \mathcal{L}(\mathbf{v}', \boldsymbol{\mu}) - \mathcal{L}(\mathbf{v}, \boldsymbol{\mu}') \\ &= \mathbf{p}^\top \mathbf{v}' + \boldsymbol{\mu}^\top \mathbf{a}_{\mathbf{v}'} - \mathbf{p}^\top \mathbf{v} - \boldsymbol{\mu}'^\top \mathbf{a}_{\mathbf{v}} \end{aligned} \quad (14)$$

which is skew-symmetric, and  $x^* := (\mathbf{v}^*, \boldsymbol{\mu}^*)$  is a solution to EP( $\mathcal{X}, F$ ). This identification gives us a COL problem with a predictable<sup>5</sup>, linear loss

$$l_n(x) := \mathbf{p}^\top \mathbf{v} + \boldsymbol{\mu}_n^\top \mathbf{a}_{\mathbf{v}} - \mathbf{p}^\top \mathbf{v}_n - \boldsymbol{\mu}^\top \mathbf{a}_{\mathbf{v}_n} \quad (15)$$

where  $x_n = (\mathbf{v}_n, \boldsymbol{\mu}_n)$ . We remark that tackling (15) with an algorithm that isolates updates for  $\mathbf{v}$  and  $\boldsymbol{\mu}$  is the same as applying a similar algorithm in the two-player setup. The merit of the COL viewpoint is mainly in the EP-inspired insights into the regret analyses.

#### 3.2 Policy Performance and Residual

By Proposition 1, any no-regret algorithm, when applied to (15), provides guarantees in terms of the residual function  $r_{ep}(x)$  of the EP. But this is not the end of the story. We also need to relate the learner decision  $x \in \mathcal{X}$  to a policy  $\pi$  in RL and then convert bounds on  $r_{ep}(x)$  back to the policy performance  $V^\pi(p)$ . Here we

<sup>5</sup>  $l_n$  can be (partially) inferred from past feedbacks, as the MDP involved in each round is the same.

follow the common rule in the literature and associate each  $x = (\mathbf{v}, \boldsymbol{\mu}) \in \mathcal{X}$  with a policy  $\pi_{\boldsymbol{\mu}}$  defined as

$$\pi_{\boldsymbol{\mu}}(a|s) \propto \mu(s, a). \quad (16)$$

In the following, we relate the residual  $r_{ep}(x)$  to the performance gap  $V^*(p) - V^{\pi_{\boldsymbol{\mu}}}(p)$  through a relative performance measure defined as

$$r_{ep}(x; x') := F(x, x) - F(x, x') = -F(x, x') \quad (17)$$

for  $x, x' \in \mathcal{X}$ , where the last equality follows from the skew-symmetry of  $F$  in (14). Intuitively, we can view  $r_{ep}(x; x')$  as comparing the performance of  $x$  with respect to the comparator  $x'$  under an optimization problem proposed by  $x$ , e.g. we have  $l_n(x_n) - l_n(x') = r_{ep}(x_n; x')$ . And by the definition in (17), it holds that  $r_{ep}(x; x') \leq \max_{x' \in \mathcal{X}} -F(x, x') = r_{ep}(x)$ .

We are looking for inequalities in the form  $V^*(p) - V^{\pi_{\boldsymbol{\mu}}}(p) \leq \kappa(r_{ep}(x; x'))$  that hold for *all*  $x \in \mathcal{X}$  with some strictly increasing function  $\kappa$  and some  $x' \in \mathcal{X}$ , so we can get *non-asymptotic* performance guarantees once we combine the two steps described at the beginning of this section. For example, directly applying results of Cheng et al. (2019a) to the COL in (15) gives  $V^*(p) - V^{\hat{\pi}_N}(p) \leq \kappa(\frac{\text{Regret}_N}{N})$ , where  $\hat{\pi}_N$  is the policy associated with the average/best decision in  $\{x_n\}_{n=1}^N$ .

### 3.2.1 The Classic Result

Existing approaches (e.g. Chen and Wang (2016); Wang (2017a); Lee and He (2018)) to the saddle-point formulation in (7) rely on the relative residual  $r_{ep}(x; x^*)$  with respect to the optimal solution to the problem  $x^*$ , which we restate in our notation.

**Proposition 2.** *For any  $x = (\mathbf{v}, \boldsymbol{\mu}) \in \mathcal{X}$ , if  $\mathbf{E}^\top \boldsymbol{\mu} \geq (1 - \gamma)\mathbf{p}$ ,  $r_{ep}(x; x^*) \geq (1 - \gamma) \min_s p(s) \|\mathbf{v}^* - \mathbf{v}^{\pi_{\boldsymbol{\mu}}}\|_\infty$ .*

Therefore, although the original saddle-point problem in (7) is framed using  $\mathcal{V}$  and  $\mathcal{M}$ , in practice, an extra constraint, such as  $\mathbf{E}^\top \boldsymbol{\mu} \geq (1 - \gamma)\mathbf{p}$ , is added into  $\mathcal{M}$ . These algorithms consider instead

$$\mathcal{M}' = \{\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} : \boldsymbol{\mu} \in \mathcal{M}, \mathbf{E}^\top \boldsymbol{\mu} \geq (1 - \gamma)\mathbf{p}\}, \quad (18)$$

so that the marginal of the estimate  $\boldsymbol{\mu}$  can have the sufficient coverage required in Proposition 2. This condition is needed to establish non-asymptotic guarantees on the performance of the policy generated by  $\boldsymbol{\mu}$  (Wang and Chen, 2016; Wang, 2017a; Lee and He, 2018), but it can sometimes be impractical to realize when  $\mathbf{p}$  is unknown as here. Without it, extra assumptions (like ergodicity in Wang (2017a)) on the MDP are needed; although it is possible to modify this constraint to use a uniform distribution instead of  $\mathbf{p}$ , this scheme worsens the constant factor and could introduce bias.

In addition, the bound in Proposition 2 is quite conservative, as it concerns the uniform error  $\|\mathbf{v}^* - \mathbf{v}^{\pi_{\boldsymbol{\mu}}}\|_\infty$

whereas the objective in RL is about the gap  $V^*(p) - V^{\pi_{\boldsymbol{\mu}}}(p) = \mathbf{p}^\top (\mathbf{v}^* - \mathbf{v}^{\pi_{\boldsymbol{\mu}}})$  with respect to the initial distribution  $p$  (i.e. a weighted error). Also, the constant term  $(1 - \gamma) \min_s p(s)$  can be quite small (e.g. when  $p$  is uniform, it is  $\frac{1 - \gamma}{|\mathcal{S}|}$ ) which can significantly amplify the error in the residual. Since a no-regret algorithm typically decreases the residual in  $O(N^{-1/2})$  after seeing  $N$  samples, a factor of  $\frac{1 - \gamma}{|\mathcal{S}|}$  would multiply the sample complexity by  $\frac{|\mathcal{S}|^2}{(1 - \gamma)^2}$ . This makes existing saddle-point approaches sample inefficient in comparison with RL methods like Q-learning (Kakade et al., 2003).

One may conjecture that the bound in Proposition 2 could perhaps be tightened by better analyses. However, we prove this is impossible in general.

**Proposition 3.** *There is a class of MDPs such that, for some  $x \in \mathcal{X}$ , Proposition 2 is an equality.*

We note that Proposition 3 does not hold for *all* MDPs. Indeed, if one makes stronger assumptions on the MDP, such as that the Markov chain induced by *every* policy is ergodic (Wang, 2017a), then it is possible to show, for all  $x \in \mathcal{X}$ ,  $r_{ep}(x; x^*) = c \|\mathbf{v}^* - \mathbf{v}^{\pi_{\boldsymbol{\mu}}}\|_\infty$  for some constant  $c$  independent of  $\gamma$  and  $|\mathcal{S}|$ , when one constrains  $\mathbf{E}^\top \boldsymbol{\mu} \geq (1 - \gamma + \gamma\sqrt{c})\mathbf{p}$ . Nonetheless, this construct still requires adding an undesirable constraint to  $\mathcal{X}$ .

### 3.2.2 Curse of Distribution Shift

Why does this happen? This issue is due to the mismatch between distributions. To better understand it, we notice a simple equality, which has often been used implicitly, e.g. in the technical proofs of Wang (2017a).

**Lemma 1.** *For any  $x = (\mathbf{v}, \boldsymbol{\mu})$ , if  $x' \in \mathcal{X}$  satisfies (2) and (5) (i.e.  $\mathbf{v}'$  and  $\boldsymbol{\mu}'$  are the value function and state-action distribution of policy  $\pi_{\boldsymbol{\mu}'}$ ),  $r_{ep}(x; x') = -\boldsymbol{\mu}^\top \mathbf{a}_{\mathbf{v}'}$ .*

Lemma 1 implies  $r_{ep}(x; x^*) = -\boldsymbol{\mu}^\top \mathbf{a}_{\mathbf{v}^*}$ , which is non-negative. This is similar to the performance difference lemma (Ng et al., 1999; Kakade and Langford, 2002).

**Lemma 2.** *Let  $\mathbf{v}^\pi$  and  $\boldsymbol{\mu}^\pi$  denote the value and state-action distribution of some policy  $\pi$ . Then for any function  $\mathbf{v}'$ , it holds that  $\mathbf{p}^\top (\mathbf{v}^\pi - \mathbf{v}') = (\boldsymbol{\mu}^\pi)^\top \mathbf{a}_{\mathbf{v}'}$ . In particular, it implies  $V^\pi(p) - V^{\pi'}(p) = (\boldsymbol{\mu}^\pi)^\top \mathbf{a}_{\mathbf{v}^{\pi'}}$ .*

From Lemmas 1 and 2, we see that the difference between the residual  $r_{ep}(x; x^*) = -\boldsymbol{\mu}^\top \mathbf{a}_{\mathbf{v}^*}$  and the performance gap  $V^{\pi_{\boldsymbol{\mu}}}(p) - V^{\pi^*}(p) = (\boldsymbol{\mu}^{\pi_{\boldsymbol{\mu}}})^\top \mathbf{a}_{\mathbf{v}^*}$  is due to the mismatch between  $\boldsymbol{\mu}$  and  $\boldsymbol{\mu}^{\pi_{\boldsymbol{\mu}}}$ , or more specifically, the mismatch between the two marginals  $\mathbf{d} = \mathbf{E}^\top \boldsymbol{\mu}$  and  $\mathbf{d}^{\pi_{\boldsymbol{\mu}}} = \mathbf{E}^\top \boldsymbol{\mu}^{\pi_{\boldsymbol{\mu}}}$ . Indeed, when  $\mathbf{d} = \mathbf{d}^{\pi_{\boldsymbol{\mu}}}$ , the residual is equal to the performance gap. However, in general, we do not have control over that difference for the sequence of variables  $\{x_n = (\mathbf{v}_n, \boldsymbol{\mu}_n) \in \mathcal{X}\}$  an algorithm generates. The sufficient condition in

Proposition 2 attempts to mitigate the difference, using the fact  $\mathbf{d}^{\pi\mu} = (1 - \gamma)\mathbf{p} + \gamma\mathbf{P}_{\pi\mu}^\top \mathbf{d}^{\pi\mu}$  from (2), where  $\mathbf{P}_{\pi\mu}$  is the transition matrix under  $\pi_\mu$ . But the missing half  $\gamma\mathbf{P}_{\pi\mu}^\top \mathbf{d}^{\pi\mu}$  (due to the long-term effects in the MDP) introduces the unavoidable, weak constant  $(1 - \gamma) \min_s p(s)$ , if we want to have a uniform bound on  $\|\mathbf{v}^* - \mathbf{v}^{\pi\mu}\|_\infty$ . The counterexample in Proposition 3 was designed to maximize the effect of distribution shift, so that  $\mu$  fails to capture state-action pairs with high advantage. To break the curse, we must properly weight the gap between  $\mathbf{v}^*$  and  $\mathbf{v}^{\pi\mu}$  instead of relying on the uniform bound on  $\|\mathbf{v}^* - \mathbf{v}^{\pi\mu}\|_\infty$  as before.

## 4 THE REDUCTION

The analyses above reveal both good and bad properties of the saddle-point setup in (7). On the one hand, we showed that approximate solutions to the saddle-point problem in (7) can be obtained by running any no-regret algorithm in the single-player COL problem defined in (15); many efficient algorithms are available from the online learning literature. On the other hand, we also discovered a root difficulty in converting an approximate solution of (7) to an approximately optimal policy in RL (Proposition 2), even after imposing strong conditions like (18). At this point, one may wonder if the formulation based on (7) is fundamentally sample inefficient compared with other approaches to RL, but this is actually not true.

Our main contribution shows that learning a policy through running a no-regret algorithm in the COL problem in (15) is, in fact, as sample efficient in policy performance as other RL techniques, even without the common constraint in (18) or extra assumptions on the MDP like ergodicity imposed in the literature.

**Theorem 1.** *Let  $X_N = \{x_n \in \mathcal{X}\}_{n=1}^N$  be any sequence. Let  $\hat{\pi}_N$  be the policy given by  $\hat{x}_N$  via (16), which is either the average or the best decision in  $X_N$ . Define  $y_N^* := (\mathbf{v}^{\hat{\pi}_N}, \mu^*)$ . Then  $V^{\hat{\pi}_N}(p) \geq V^*(p) - \frac{\text{Regret}_N(y_N^*)}{N}$ .*

Theorem 1 shows that if  $X_N$  has sublinear regret, then both the average policy and the best policy in  $X_N$  converge to the optimal policy in performance with a rate  $O(\text{Regret}_N(y_N^*)/N)$ . Compared with existing results obtained through Proposition 2, the above result removes the factor  $(1 - \gamma) \min_s p(s)$  and imposes *no* assumption on  $X_N$  or the MDP. Indeed Theorem 1 holds for *any* sequence. For example, when  $X_N$  is generated by stochastic feedback of  $l_n$ , Theorem 1 continues to hold, as the regret is defined in terms of  $l_n$ , not of the sampled loss. Stochasticity only affects the regret rate.

In other words, we have shown that when  $\mu$  and  $\mathbf{v}$  can be directly parameterized, an approximately optimal policy for the RL problem can be obtained by running any no-regret online learning algorithm, and

that the policy quality is simply dictated by the regret rate. To illustrate, in Section 5 we will prove that simply running mirror descent in this COL produces an RL algorithm that is as sample efficient as other common RL techniques. One can further foresee that algorithms leveraging the continuity in COL—e.g. mirror-prox (Juditsky et al., 2011) or PicCoLO (Cheng et al., 2019b)—and variance reduction can lead to more sample efficient RL algorithms.

Below we will also demonstrate how to use the fact that COL is *single-player* (see Section 2.3) to cleanly incorporate the effects of using function approximators to model  $\mu$  and  $\mathbf{v}$ . We will present a corollary of Theorem 1, which separates the problem of *learning*  $\mu$  and  $\mathbf{v}$ , and that of *approximating*  $\mathcal{M}$  and  $\mathcal{V}$  with function approximators. The first part is controlled by the rate of regret in online learning, and the second part depends on only the chosen class of function approximators, independently of the learning process. As these properties are agnostic to problem setups and algorithms, our reduction leads to a framework for systematic synthesis of new RL algorithms with performance guarantees. The missing proofs of this section are in Appendix C.

### 4.1 Proof of Theorem 1

The main insight of our reduction is to adopt, in defining  $r_{ep}(x; x')$ , a comparator  $x' \in \mathcal{X}$  based on the output of the algorithm (represented by  $x$ ), instead of the fixed comparator  $x^*$  (the optimal pair of value function and state-action distribution) that has been used conventionally, e.g. in Proposition 2. While this idea seems unnatural from the standard saddle-point or EP perspective, it is possible, because the regret in online learning is measured against the worst-case choice in  $\mathcal{X}$ , which is allowed to be selected in *hindsight*. Specifically, we propose to select the following comparator to directly bound  $V^*(p) - V^{\hat{\pi}_N}(p)$  instead of the conservative measure  $\|V^* - V^{\hat{\pi}_N}\|_\infty$  used before.

**Proposition 4.** *For  $x = (\mathbf{v}, \mu) \in \mathcal{X}$ , define  $y_x^* := (\mathbf{v}^{\pi_\mu}, \mu^*) \in \mathcal{X}$ . It holds  $r_{ep}(x; y_x^*) = V^*(p) - V^{\pi_\mu}(p)$ .*

To finish the proof, let  $\hat{x}_N$  be either  $\frac{1}{N} \sum_{n=1}^N x_n$  or  $\arg \min_{x \in X_N} r_{ep}(x; y_x^*)$ , and let  $\hat{\pi}_N$  denote the policy given by (16). First,  $V^*(p) - V^{\hat{\pi}_N}(p) = r_{ep}(\hat{x}_N; y_N^*)$  by Proposition 4. Next we follow the proof idea of Proposition 1 in Cheng et al. (2019a): since  $F$  is skew-symmetric and  $F(y_N^*, \cdot)$  is convex, we have by (17)

$$\begin{aligned} V^*(p) - V^{\hat{\pi}_N}(p) &= r_{ep}(\hat{x}_N; y_N^*) = -F(\hat{x}_N, y_N^*) \\ &= F(y_N^*, \hat{x}_N) \leq \frac{1}{N} \sum_{n=1}^N F(y_N^*, x_n) \\ &= \frac{1}{N} \sum_{n=1}^N -F(x_n, y_N^*) = \frac{1}{N} \text{Regret}_N(y_N^*). \end{aligned}$$

## 4.2 Function Approximators

When the state and action spaces are large or continuous, directly optimizing  $\mathbf{v}$  and  $\boldsymbol{\mu}$  can be impractical. Instead we can consider optimizing over a subset of feasible choices parameterized by function approximators

$$\mathcal{X}_\Theta = \{\mathbf{x}_\theta = (\phi_\theta, \psi_\theta) : \psi_\theta \in \mathcal{M}, \theta \in \Theta\}, \quad (19)$$

where  $\phi_\theta$  and  $\psi_\theta$  are functions parameterized by  $\theta \in \Theta$ , and  $\Theta$  is a parameter set. Because COL is a single-player setup, we can extend the previous idea and Theorem 1 to provide performance bounds in this case by a simple rearrangement (see Appendix C), which is a common trick used in the online imitation learning literature (Ross et al., 2011; Cheng and Boots, 2018; Cheng et al., 2019c). Notice that, in (19), we require only  $\psi_\theta \in \mathcal{M}$ , but not  $\phi_\theta \in \mathcal{V}$ , because for the performance bound in our reduction to hold, we only need the constraint  $\mathcal{M}$  (see Lemma 4 in proof of Proposition 4).

**Corollary 1.** *Let  $X_N = \{x_n \in \mathcal{X}_\theta\}_{n=1}^N$  be any sequence. Let  $\hat{\pi}_N$  be the policy given either by the average or the best decision in  $X_N$ . It holds that*

$$V^{\hat{\pi}_N}(p) \geq V^*(p) - \frac{\text{Regret}_N(\Theta)}{N} - \epsilon_{\Theta, N}$$

where  $\epsilon_{\Theta, N} = \min_{x_\theta \in \mathcal{X}_\theta} r_{ep}(\hat{x}_N; y_N^*) - r_{ep}(\hat{x}_N; x_\theta)$  measures the expressiveness of  $X_\theta$ , and  $\text{Regret}_N(\Theta) := \sum_{n=1}^N l_n(x_n) - \min_{x \in \mathcal{X}_\Theta} \sum_{n=1}^N l_n(x)$ .

We can quantify  $\epsilon_{\Theta, N}$  with the basic Hölder's inequality.

**Proposition 5.** *Let  $\hat{x}_N = (\hat{\mathbf{v}}_N, \hat{\boldsymbol{\mu}}_N)$ . Under the setup in Corollary 1, regardless of the parameterization, it is true that  $\epsilon_{\Theta, N}$  is no larger than*

$$\begin{aligned} & \min_{(\mathbf{v}_\theta, \boldsymbol{\mu}_\theta) \in \mathcal{X}_\Theta} \frac{\|\boldsymbol{\mu}_\theta - \boldsymbol{\mu}^*\|_1}{1 - \gamma} + \min_{\mathbf{w}: \mathbf{w} \geq \mathbf{1}} \|\mathbf{b}_{\hat{\boldsymbol{\mu}}_N}\|_{1, \mathbf{w}} \|\mathbf{v}_\theta - \mathbf{v}^{\hat{\pi}_N}\|_{\infty, 1/\mathbf{w}} \\ & \leq \min_{(\mathbf{v}_\theta, \boldsymbol{\mu}_\theta) \in \mathcal{X}_\Theta} \frac{1}{1 - \gamma} \left( \|\boldsymbol{\mu}_\theta - \boldsymbol{\mu}^*\|_1 + 2\|\mathbf{v}_\theta - \mathbf{v}^{\hat{\pi}_N}\|_{\infty} \right). \end{aligned}$$

where the norms are defined as  $\|\mathbf{x}\|_{1, \mathbf{w}} = \sum_i w_i |x_i|$  and  $\|\mathbf{x}\|_{\infty, 1/\mathbf{w}} = \max_i w_i^{-1} |x_i|$ .

Proposition 5 says  $\epsilon_{\Theta, N}$  depends on how well  $\mathcal{X}_\Theta$  captures the value function of the output policy  $\mathbf{v}^{\hat{\pi}_N}$  and the optimal state-action distribution  $\boldsymbol{\mu}^*$ . We remark that this result is independent of how  $\mathbf{v}^{\hat{\pi}_N}$  is generated. Furthermore, Proposition 5 makes *no* assumption whatsoever on the structure of function approximators. It even allows sharing parameters  $\theta$  between  $\mathbf{v} = \phi_\theta$  and  $\boldsymbol{\mu} = \psi_\theta$ , e.g., they can be a bi-headed neural network, which is common for learning shared feature representations. More precisely, the structure of the function approximator would only affect whether  $l_n((\phi_\theta, \psi_\theta))$  remains a convex function in  $\theta$ , which determines the difficulty of designing algorithms with sublinear regret.

In summary, the proposed COL formulation provides a reduction which dictates the policy performance with

---

## Algorithm 1 Mirror descent for RL

---

**Input:**  $\epsilon$  optimality of the  $\gamma$ -average return  
 $\delta$  maximal failure probability  
 generative model of an MDP

**Output:**  $\hat{\pi}_N = \pi^{\hat{\boldsymbol{\mu}}_N}$

- 1:  $x_1 = (\mathbf{v}_1, \boldsymbol{\mu}_1)$  where  $\boldsymbol{\mu}_1$  is uniform and  $\mathbf{v}_1 \in \mathcal{V}$
  - 2: Set  $N = \tilde{\Omega}\left(\frac{|\mathcal{S}||\mathcal{A}| \log(\frac{1}{\delta})}{(1-\gamma)^2 \epsilon^2}\right)$  and  $\eta = (1-\gamma)(|\mathcal{S}||\mathcal{A}|N)^{-1/2}$
  - 3: Set the Bregman divergence as (21)
  - 4: **for**  $n = 1 \dots N - 1$  **do**
  - 5:   Sample  $g_n$  according to (23)
  - 6:   Update  $x_{n+1}$  according to (20)
  - 7: **end for**
  - 8: Set  $(\hat{\mathbf{v}}_N, \hat{\boldsymbol{\mu}}_N) = \hat{x}_N = \frac{1}{N} \sum_{n=1}^N x_n$
- 

two separate factors: 1) the rate of regret  $\text{Regret}_N(\Theta)$  which is controlled by the choice of online learning algorithm; 2) the approximation error  $\epsilon_{\Theta, N}$  which is determined by the choice of function approximators. These two factors can almost be treated independently, except that the choice of function approximators would determine the properties of  $l_n((\phi_\theta, \psi_\theta))$  as a function of  $\theta$ , and the choice of  $\Theta$  needs to ensure (19) is admissible.

## 5 SAMPLE COMPLEXITY OF MIRROR DESCENT

We demonstrate the power of our reduction by applying perhaps the simplest online learning algorithm, mirror descent, to the proposed COL problem in (15) with stochastic feedback (Algorithm 1). For transparency, we first discuss the tabular setup. We will show a natural extension to basis functions at the end. Recall that mirror descent is a first-order algorithm, whose update rule can be written as

$$x_{n+1} = \arg \min_{x \in \mathcal{X}} \langle g_n, x \rangle + \frac{1}{\eta} B_R(x|x_n) \quad (20)$$

where  $\eta > 0$  is the step size,  $g_n$  is the feedback direction, and  $B_R(x|x') = R(x) - R(x') - \langle \nabla R(x'), x - x' \rangle$  is the Bregman divergence generated by a strictly convex function  $R$ . Based on the geometry of  $\mathcal{X} = \mathcal{V} \times \mathcal{M}$ , we consider a natural Bregman divergence of the form

$$B_R(x'|x) = \frac{1}{2|\mathcal{S}|} \|\mathbf{v}' - \mathbf{v}\|_2^2 + KL(\boldsymbol{\mu}'|\boldsymbol{\mu}) \quad (21)$$

This choice mitigates the effects of dimension (e.g. if we set  $x_1 = (\mathbf{v}_1, \boldsymbol{\mu}_1)$  with  $\boldsymbol{\mu}_1$  being the uniform distribution, it holds  $B_R(x'|x_1) = \tilde{O}(1)$  for any  $x' \in \mathcal{X}$ ).

To define the feedback direction  $g_n$ , we slightly modify the per-round loss  $l_n$  in (15) and consider a new loss

$$h_n(x) := \mathbf{b}_{\boldsymbol{\mu}_n}^\top \mathbf{v} + \boldsymbol{\mu}^\top \left( \frac{1}{1-\gamma} \mathbf{1} - \mathbf{a}_{\mathbf{v}_n} \right) \quad (22)$$

that shifts  $l_n$  by a constant, where  $\mathbf{1}$  is the vector of ones. One can verify that  $l_n(x) - l_n(x') = h_n(x) - h_n(x')$ , for all  $x, x' \in \mathcal{X}$ . Therefore, using  $h_n$  does not change regret. The reason for using  $h_n$  instead of  $l_n$  is to

make  $\nabla_{\boldsymbol{\mu}} h_n((\mathbf{v}, \boldsymbol{\mu}))$  (and its unbiased approximation) a positive vector, so the regret bound can have a better dimension dependency. This is a common trick used in online learning (e.g. in EXP3 Hazan (2016)) for optimizing variables living in a simplex ( $\boldsymbol{\mu}$  here).

We set the first-order feedback  $g_n$  as an unbiased *sampled* estimate of  $\nabla h_n(x_n)$ . In round  $n$ , this is realized by two independent calls of the generative model:

$$g_n = \left[ \begin{array}{c} \tilde{\mathbf{P}}_n + \frac{1}{1-\gamma}(\gamma\tilde{\mathbf{P}}_n - \mathbf{E}_n)^\top \tilde{\boldsymbol{\mu}}_n \\ |\mathcal{S}||\mathcal{A}| \left( \frac{1}{1-\gamma} \hat{\mathbf{1}}_n - \hat{\mathbf{r}}_n - \frac{1}{1-\gamma}(\gamma\hat{\mathbf{P}}_n - \hat{\mathbf{E}}_n) \mathbf{v}_n \right) \end{array} \right] \quad (23)$$

Let  $g_n = [\mathbf{g}_{n,v}; \mathbf{g}_{n,\mu}]$ . For  $\mathbf{g}_{n,v}$ , we sample  $\mathbf{p}$ , sample  $\boldsymbol{\mu}_n$  to get a state-action pair, and query the transition  $\mathbf{P}$  at the state-action pair sampled from  $\boldsymbol{\mu}_n$ . ( $\tilde{\mathbf{P}}_n$ ,  $\tilde{\mathbf{P}}_n$ , and  $\tilde{\boldsymbol{\mu}}_n$  denote the single-sample estimate of these probabilities.) For  $\mathbf{g}_{n,\mu}$ , we first sample *uniformly* a state-action pair (which explains the factor  $|\mathcal{S}||\mathcal{A}|$ ), and then query the reward  $\mathbf{r}$  and the transition  $\mathbf{P}$ . ( $\hat{\mathbf{1}}_n$ ,  $\hat{\mathbf{r}}_n$ ,  $\hat{\mathbf{P}}_n$ , and  $\hat{\mathbf{E}}_n$  denote the single-sample estimates.) To emphasize, we use  $\tilde{\cdot}$  and  $\hat{\cdot}$  to distinguish the empirical quantities obtained by these two independent queries. By construction, we have  $\mathbf{g}_{n,\mu} \geq 0$ . It is clear that this direction  $g_n$  is unbiased, i.e.  $\mathbb{E}[g_n] = \nabla h_n(x_n)$ . Moreover, it is extremely sparse and can be computed using  $O(1)$  sample, computational, and memory complexities.

Below we show this algorithm, despite being extremely simple, has strong theoretical guarantees. In other words, we obtain simpler versions of the algorithms proposed in Wang and Chen (2016); Wang (2017a); Chen et al. (2018), but with improved performance.

**Theorem 2.** *With probability  $1 - \delta$ , Algorithm 1 learns an  $\epsilon$ -optimal policy with  $\tilde{O}\left(\frac{|\mathcal{S}||\mathcal{A}| \log(\frac{1}{\delta})}{(1-\gamma)^2 \epsilon^2}\right)$  samples.*

Note that the above statement makes no assumption on the MDP (except the tabular setup for simplifying analysis). Also, because the definition of value function in (1) is scaled by a factor  $(1 - \gamma)$ , the above result translates into a sample complexity in  $\tilde{O}\left(\frac{|\mathcal{S}||\mathcal{A}| \log(\frac{1}{\delta})}{(1-\gamma)^4 \epsilon^2}\right)$  for the conventional discounted accumulated rewards.

### 5.1 Proof Sketch of Theorem 2

The proof is based on basic properties of mirror descent and martingale concentration (see Appendix D for the full details). Let  $y_N^* = (\mathbf{v}^{\hat{\pi}_N}, \boldsymbol{\mu}^*)$ . We bound the regret in Theorem 1 using the facts that  $h_n$  is a constant shift from  $l_n$  and that  $h_n(\cdot)$  is linear.

$$\begin{aligned} \text{Regret}_N(y_N^*) &\leq \left( \sum_{n=1}^N (\nabla h_n(x_n) - g_n)^\top x_n \right) \\ &+ \left( \max_{x \in \mathcal{X}} \sum_{n=1}^N g_n^\top (x_n - x) \right) + \left( \sum_{n=1}^N (g_n - \nabla h_n(x_n))^\top y_N^* \right) \end{aligned}$$

The first term is a martingale ( $x_n$  does not depend on  $g_n$ ), we apply a Bernstein-type martingale concentra-

tion to it and show it is in  $\tilde{O}\left(\frac{\sqrt{N|\mathcal{S}||\mathcal{A}| \log(\frac{1}{\delta})}}{1-\gamma}\right)$ . For the second term, we treat  $g_n^\top x$  as the per-round loss and use standard regret analysis of mirror descent to show a bound in  $\tilde{O}\left(\frac{\sqrt{N|\mathcal{S}||\mathcal{A}|}}{1-\gamma}\right)$ . The third term is *not* a martingale ( $\mathbf{v}^{\hat{\pi}_N}$  in  $y_N^* = (\mathbf{v}^{\hat{\pi}_N}, \boldsymbol{\mu}^*)$  depends on  $\{g_n\}_{n=1}^N$ ). Using a union bound, we show it is again no more than  $\tilde{O}\left(\frac{\sqrt{N|\mathcal{S}||\mathcal{A}| \log(\frac{1}{\delta})}}{1-\gamma}\right)$ . The union bound does not increase the rate because we only need to handle  $\mathbf{v}^{\hat{\pi}_N}$ , not  $\boldsymbol{\mu}^*$  which induces a martingale. To finish the proof, we substitute this h-p regret bound into Theorem 1.

### 5.2 Extension to Function Approximators

The above algorithm assumes a tabular setup for illustration purposes. In Appendix E, we describe a direct extension of Algorithm 1 that uses linearly parameterized function approximators of the form  $x_\theta = (\boldsymbol{\Phi}\boldsymbol{\theta}_v, \boldsymbol{\Psi}\boldsymbol{\theta}_\mu)$ , where columns of bases  $\boldsymbol{\Phi}, \boldsymbol{\Psi}$  belong to  $\mathcal{V}$  and  $\mathcal{M}$ , respectively, and  $(\boldsymbol{\theta}_v, \boldsymbol{\theta}_\mu) \in \Theta$ . The overall algorithm stays the same, but the gradient is computed by chain-rule. This can be done in  $O(\dim(\Theta))$  time and space (compared to the slow  $O(|\mathcal{S}||\mathcal{A}|)$  required before for the projection in (20)), as we now only optimize in  $\Theta$ . We also prove that the sample complexity is better, though at the cost of bias  $\epsilon_{\Theta,N}$  in Corollary 1. Therefore, the algorithm becomes applicable to large-scale or continuous problems.

**Theorem 3.** *Under a proper choice of  $\Theta$  and  $B_R$ , with probability  $1 - \delta$ , Algorithm 1 learns an  $(\epsilon + \epsilon_{\Theta,N})$ -optimal policy with  $\tilde{O}\left(\frac{\dim(\Theta) \log(\frac{1}{\delta})}{(1-\gamma)^2 \epsilon^2}\right)$  samples.*

The full proof is in Appendix E. In short, for a  $\Theta$  satisfying (19), we use Corollary 1 to reduce the problem into regret minimization. We then sample uniformly over the  $\dim(\Theta)$  columns of  $\boldsymbol{\Psi}$  (instead of over all states and actions like (23)) when computing unbiased estimates of  $\nabla_{\boldsymbol{\theta}_\mu} h_n((\boldsymbol{\theta}_v, \boldsymbol{\theta}_\mu))$ .

## 6 CONCLUSION

We propose a reduction from RL to no-regret online learning that provides a systematic way to design new RL algorithms with performance guarantees. Compared with existing approaches, our framework makes no assumption on the MDP and naturally works with function approximators. To illustrate, we design a simple RL algorithm based on mirror descent; it achieves similar sample complexity as other RL techniques and is scalable to large or continuous problems. This encouraging result shows the strength of the online learning perspective. As future work, we think faster learning in RL is possible by using control variates for variance reduction or applying advanced online techniques (Rakhlin and Sridharan, 2012; Cheng et al., 2019b) that exploit the continuity in COL to predict future gradients.



## Acknowledgements

This research is partially supported by NVIDIA Graduate Fellowship.

## References

- Alan S Manne et al. Linear programming and sequential decision models. Technical report, Cowles Foundation for Research in Economics, Yale University, 1959.
- Eric V Denardo and Bennett L Fox. Multichain Markov renewal programs. *SIAM Journal on Applied Mathematics*, 16(3):468–487, 1968.
- Mengdi Wang and Yichen Chen. An online primal-dual method for discounted Markov decision processes. In *Conference on Decision and Control*, pages 4516–4521. IEEE, 2016.
- Yichen Chen and Mengdi Wang. Stochastic primal-dual methods and sample complexity of reinforcement learning. *arXiv preprint arXiv:1612.02516*, 2016.
- Mengdi Wang. Randomized linear programming solves the discounted Markov decision problem in nearly-linear running time. *ArXiv e-prints*, 2017a.
- Donghwan Lee and Niao He. Stochastic primal-dual Q-learning. *arXiv preprint arXiv:1810.08298*, 2018.
- Mengdi Wang. Primal-dual  $\pi$  learning: Sample complexity and sublinear run time for ergodic Markov decision problems. *arXiv preprint arXiv:1710.06100*, 2017b.
- Qihang Lin, Selvaprabu Nadarajah, and Negar Soheili. Revisiting approximate linear programming using a saddle point based reformulation and root finding solution approach. Technical report, working paper, U. of Il. at Chicago and U. of Iowa, 2017.
- Bo Dai, Albert Shaw, Niao He, Lihong Li, and Le Song. Boosting the actor with dual critic. In *International Conference on Learning Representation*, 2018.
- Yichen Chen, Lihong Li, and Mengdi Wang. Scalable bilinear  $\pi$  learning using state and action features. *arXiv preprint arXiv:1804.10328*, 2018.
- Chandrashekar Lakshminarayanan, Shalabh Bhatnagar, and Csaba Szepesvári. A linearly relaxed approximate linear program for Markov decision processes. *IEEE Transactions on Automatic Control*, 63(4):1185–1191, 2018.
- Jyrki Kivinen and Manfred K Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *information and computation*, 132(1):1–63, 1997.
- Geoffrey J Gordon. Regret bounds for prediction problems. In *Conference on Learning Theory*, volume 99, pages 29–40, 1999.
- Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning*, pages 928–936, 2003.
- Ching-An Cheng, Jonathan Lee, Ken Goldberg, and Byron Boots. Online learning with continuous variations: Dynamic regret and reductions. *arXiv preprint arXiv:1902.07286*, 2019a.
- Eugen Blum. From optimization and variational inequalities to equilibrium problems. *Math. student*, 63:123–145, 1994.
- M Bianchi and S Schaible. Generalized monotone bifunctions and equilibrium problems. *Journal of Optimization Theory and Applications*, 90(1):31–43, 1996.
- Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. *arXiv preprint arXiv:1208.3728*, 2012.
- Ching-An Cheng, Xinyan Yan, Nathan Ratliff, and Byron Boots. Predictor-corrector policy optimization. In *International Conference on Machine Learning*, 2019b.
- Michael J Kearns and Satinder P Singh. Finite-sample convergence rates for q-learning and indirect algorithms. In *Advances in neural information processing systems*, pages 996–1002, 1999.
- Sham Machandranath Kakade et al. *On the sample complexity of reinforcement learning*. PhD thesis, University of London London, England, 2003.
- Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.
- Onésimo Hernández-Lerma and Jean B Lasserre. *Discrete-time Markov control processes: basic optimality criteria*, volume 30. Springer Science & Business Media, 2012.
- Richard Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–515, 1954.
- Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.
- Anatoli Juditsky, Arkadi Nemirovski, and Claire Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1):17–58, 2011.
- Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.
- Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

- Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.
- Alejandro Jofré and Roger J-B Wets. Variational convergence of bifunctions: motivating applications. *SIAM Journal on Optimization*, 24(4):1952–1979, 2014.
- Jacob Abernethy, Peter L Bartlett, and Elad Hazan. Blackwell approachability and no-regret learning are equivalent. In *Annual Conference on Learning Theory*, pages 27–46, 2011.
- Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning*, volume 99, pages 278–287, 1999.
- Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, volume 2, pages 267–274, 2002.
- Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, pages 627–635, 2011.
- Ching-An Cheng and Byron Boots. Convergence of value aggregation for imitation learning. *International Conference on Artificial Intelligence and Statistics*, 2018.
- Ching-An Cheng, Xinyan Yan, Evangelos A Theodorou, and Byron Boots. Accelerating imitation learning with predictive models. In *International Conference on Artificial Intelligence and Statistics*, 2019c.
- Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016. URL <http://dblp.uni-trier.de/db/journals/ftopt/ftopt2.html#Hazan16>.
- Colin McDiarmid. Concentration. In *Probabilistic methods for algorithmic discrete mathematics*, pages 195–248. Springer, 1998.