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 γ-discounted tabular RL problem, with probability at least $1-\delta$, it learns an ϵ-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 [1, 2] 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 [3–11]. 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 [12–14], 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 Π satisfying $V^{\hat{\pi}_N}(p) \geq V^{\pi^*}(p) - o(1) - \epsilon_\Pi$, where $V^{\pi}(p)$ is the accumulated reward of policy π with respect to some unknown initial state distribution p, π^* is the optimal policy, and $\epsilon_\Pi \geq 0$ is a measure of the expressivity of Π (see Section 4.2 for definition).

Our reduction is built on a refinement of online learning, called Continuous Online Learning (COL) [15], 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) [16, 17], and any monotone EP (including our saddle-point problem of interest) can be framed as no-regret learning in a properly constructed COL problem [15]. 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 *in-*

Proceedings of the 23$^{\text{rd}}$International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108. Copyright 2020 by the author(s).

dependently of the learning process. For example, one can accelerate learning by adopting optimistic online algorithms [18, 19] 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[1] we first prove that, for *any* tabular Markov decision process (MDP), with probability at least $1-\delta$, this algorithm learns an ϵ-optimal policy for the γ-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 γ 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 [3–6] by a large factor of $\frac{|\mathcal{S}|^2}{(1-\gamma)^2}$, *without* making any assumption on the MDP.[2] 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. [5], achieving the same rate with our new analysis, we remark that our algorithm is considerably simpler and removes a projection required in previous work [3–6].

Finally, we do note that the same sample complexity can also be achieved, e.g., by model-based RL and (phased) Q-learning [20, 21]. However, these methods either have super-linear runtime, with no obvious route for improvement, or could become unstable when using function approximators without further assumption.

2 SETUP & PRELIMINARIES

Let $\mathcal{S}$ and $\mathcal{A}$ be state and action spaces, which can be discrete or continuous. We consider γ-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 $V^\pi(p) \coloneqq \mathbb{E}_{s\sim p}[V^\pi(s)]$, where p is the initial state distribution,

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

is the value function of π at state s, $r: \mathcal{S}\times\mathcal{A}\to[0,1]$ is the reward function, and $\rho^\pi(s)$ is the distribution of trajectory $\xi = s_0, a_0, s_1, \dots$ generated by running π 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 $r(s,a)$ in the MDP 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^π in (1) contains a $(1-\gamma)$ factor. We adopt this setup to make writing more compact. We denote the optimal policy as π^* 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 [22] for details).

To show how $\max_\pi V^\pi(p)$ can be framed as a LP, let us define the average state distribution under π, $d^\pi(s) \coloneqq (1-\gamma)\sum_{t=0}^\infty \gamma^t d_t^\pi(s)$, where d_t^π is the state distribution at time t generated by running π from p (e.g. $d_0^\pi = p$). By construction, d^π 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^π, we can write $V^\pi(p) = \mathbb{E}_{s\sim d^\pi}\mathbb{E}_{a\sim\pi|s}\left[r(s,a)\right]$ and our objective $\max_\pi V^\pi(p)$ equivalently as:

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

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 or even continuous spaces (under proper regularity assumptions; see [23]). We adopt this abuse of notation (emphasized by boldfaced symbols) 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 π^*.

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

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

[1] In practice, a generative model can be approximated by running a behavior policy with sufficient exploration [20].

[2] [5] has the same sample complexity but requires the MDP to be ergodic under any policy.

[3] Our setup in (4) differs from the classic one in the $(1-\gamma)$ factor in the constraint due to the average setup.

It can be verified that for all $\mathbf{p} > 0$, the solution to (4) satisfies the Bellman equation [24] and therefore is the optimal value function $\mathbf{v}^*$ (the vector form of V^*). We note that, for any policy π, V^π by definition satisfies a stationarity property

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

which can be viewed as a dual equivalent of (2) for d^π. 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. When only generative models are available (as in our setup), one can alternatively exploit the duality relationship between the two LPs in (3) and (4), and frame RL as a saddle-point problem [3]. Let us define

$$\mathbf{a}_{\mathbf{v}} \coloneqq \mathbf{r} + \tfrac{1}{1-\gamma}(\gamma\mathbf{P} - \mathbf{E})\mathbf{v} \tag{6}$$

as the *advantage function* with respect to $\mathbf{v}$ (where $\mathbf{v}$ is not necessarily a value function). Then the Lagrangian connecting the two LPs can be written as

$$\mathcal{L}(\mathbf{v}, \boldsymbol{\mu}) \coloneqq \mathbf{p}^\top\mathbf{v} + \boldsymbol{\mu}^\top\mathbf{a}_{\mathbf{v}}, \tag{7}$$

which leads to the saddle-point formulation,

$$\min_{\mathbf{v}\in\mathcal{V}}\max_{\boldsymbol{\mu}\in\mathcal{M}} \mathcal{L}(\mathbf{v}, \boldsymbol{\mu}), \tag{8}$$

where the constraints are

$$\mathcal{V} = \{\mathbf{v} \in \mathbb{R}^{|\mathcal{S}|} : \mathbf{v} \geq 0, \|\mathbf{v}\|_\infty \leq 1\} \tag{9}$$

$$\mathcal{M} = \{\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} : \boldsymbol{\mu} \geq 0, \|\boldsymbol{\mu}\|_1 = 1\}. \tag{10}$$

The solution to (8) is exactly $(\mathbf{v}^*, \boldsymbol{\mu}^*)$, but notice that extra constraints on the norm of $\boldsymbol{\mu}$ and $\mathbf{v}$ are introduced in $\mathcal{V}, \mathcal{M}$, compared with (3) and (4). This is a common practice, which uses known bounds on the solutions of (3) and (4) (discussed above) to make the search spaces $\mathcal{V}$ and $\mathcal{M}$ in (8) compact and as small as possible, so that the optimization converges faster.

Having compact variable sets allows using first-order stochastic methods, such as stochastic mirror descent and mirror-prox [25, 26], to efficiently solve the problem. These methods only require using the generative model to 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

$$\mathbf{b}_{\boldsymbol{\mu}} \coloneqq \mathbf{p} + \tfrac{1}{1-\gamma}(\gamma\mathbf{P} - \mathbf{E})^\top\boldsymbol{\mu} \tag{11}$$

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 $\boldsymbol{\mu}$, which are often realized by linear basis functions for the sake of analysis [10].

2.3 COL and EPs

Finally, we review the COL setup in [15], which we will use to design the reduction from the saddle-point problem in (8) to online learning in the next section.

Recall that 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 $l_n : \mathcal{X} \to \mathbb{R}$ 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 $x' \in \mathcal{X}$,

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

When l_n is convex and $\mathcal{X}$ is compact and convex, many no-regret (i.e. $\mathrm{Regret}_N(x') = o(N)$) algorithms are available, such as mirror descent and follow-the-regularized-leader [27–29].

COL is a subclass of online learning problems where the loss sequence changes continuously with respect to the played decisions of the learner [15]. 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

$$l_n(x) = f_{x_n}(x), \tag{12}$$

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 [15], 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 (8) 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 $F(x,x) \geq 0$.[4] The problem $\mathrm{EP}(\mathcal{X}, F)$ aims to find $x^\star \in \mathcal{X}$ s.t.

$$F(x^\star, x) \geq 0, \qquad \forall x \in \mathcal{X}. \tag{13}$$

By its definition, a natural residual function to quantify the quality of an approximation solution x to EP is $r_{ep}(x) \coloneqq -\min_{x'\in\mathcal{X}} F(x, x')$ which describes the degree to which (13) 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 convex optimization, saddle-point problems, variational inequalities, etc. For instance, a convex-concave problem

[4] We restrict ourselves to this convex and continuous case as it is sufficient for our problem setup.

$\min_{y\in\mathcal{Y}}\max_{z\in\mathcal{Z}}\phi(y,z)$ can be cast as EP$(\mathcal{X},F)$ with $\mathcal{X}=\mathcal{Y}\times\mathcal{Z}$ and the skew-symmetric bifunction [30]

$$F(x,x') \coloneqq \phi(y',z)-\phi(y,z'), \tag{14}$$

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 [15], 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 (14)), one can construct a COL problem by setting $f_{x'}(x)\coloneqq F(x',x)$, 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. [15] *If F is* skew-symmetric *and $l_n(x)=F(x_n,x)$, then $r_{ep}(\hat{x}_N)\leq\frac{1}{N}Regret_N$, where $Regret_N=\max_{x\in\mathcal{X}}Regret_N(x)$, 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 (8). This analysis paves a way for of 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 [31]. Likewise, the relationship between the approximate solution of (8) and policy performance is also available; this is how the saddle-point formulation [5] 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 (8) as an adversarial two-player online learning problem [31], we adopt the recent reduction to COL [15] 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 is in some sense cleaner and 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 (8) 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 $\frac{|\mathcal{S}|^2}{(1-\gamma)^2}$ in sample complexity compared with the best bounds in the literature of the saddle-point setup, *without* adding extra constraints on $\mathcal{X}$ and 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 (8). 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 (14), the bifunction F of the associated EP$(\mathcal{X},F)$ is naturally given as

$$\begin{aligned} F(x,x') &\coloneqq \mathcal{L}(\mathbf{v}',\boldsymbol{\mu})-\mathcal{L}(\mathbf{v},\boldsymbol{\mu}') \\ &= \mathbf{p}^\top\mathbf{v}'+\boldsymbol{\mu}^\top\mathbf{a}_{\mathbf{v}'}-\mathbf{p}^\top\mathbf{v}-\boldsymbol{\mu}'^\top\mathbf{a}_{\mathbf{v}} \end{aligned} \tag{15}$$

which is skew-symmetric, and $x^*\coloneqq(\boldsymbol{v}^*,\boldsymbol{\mu}^*)$ is a solution to EP$(\mathcal{X},F)$. This identification gives us a COL problem with a predictable[5], linear loss

$$l_n(x)\coloneqq \mathbf{p}^\top\mathbf{v}+\boldsymbol{\mu}_n^\top\mathbf{a}_{\mathbf{v}}-\mathbf{p}^\top\mathbf{v}_n-\boldsymbol{\mu}^\top\mathbf{a}_{\mathbf{v}_n} \tag{16}$$

where $x_n=(\mathbf{v}_n,\boldsymbol{\mu}_n)$. We remark that approaching (16) with an algorithm that isolates updates for $\mathbf{v}$ and $\boldsymbol{\mu}$ is the same as applying a similar algorithm in the classic two-player setup. The merit of the COL viewpoint is mainly in the EP-inspired insight into the regret analyses.

3.2 Policy Performance and Residual

By Proposition 1, any no-regret algorithm, when applied to (16), 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 π in RL and then convert bounds on $r_{ep}(x)$ back to the policy performance $V^\pi(p)$. Here we 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

$$\pi_{\boldsymbol{\mu}}(a|s)\propto\mu(s,a). \tag{17}$$

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

$$r_{ep}(x;x')\coloneqq F(x,x)-F(x,x')=-F(x,x') \tag{18}$$

[5] l_n can be (partially) inferred from past feedbacks, as the MDP involved in each round is the same.

for $x, x' \in \mathcal{X}$, where the last equality follows from the skew-symmetry of F in (15). 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 (18), 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 κ 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, by directly applying results of [15] to the COL in (16), we get $V^*(p) - V^{\hat{\pi}_N}(p) \leq \kappa(\frac{\text{Regret}_N}{N})$, where $\hat{\pi}_N$ is the policy associated with the average/best decision in $\{x_n\}_{1=n}^N$.

3.2.1 The Classic Result

Existing approaches (e.g. [4–6]) to the saddle-point point formulation in (8) 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 (8) 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 $\mathcal{M}$, i.e. these algorithms consider instead

$$\mathcal{M}' = \{\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} : \boldsymbol{\mu} \in \mathcal{M}, \mathbf{E}^\top \boldsymbol{\mu} \geq (1-\gamma)\mathbf{p}\}, \quad (19)$$

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 $\boldsymbol{\mu}$ [3, 5, 6], but it can sometimes be impractical to realize when $\mathbf{p}$ is unknown as here. Without it, extra assumptions (like ergodicity [5]) 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, Proposition 2 is undesirable for a number of reasons. First, the bound 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). Second, 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. Because a no-regret algorithm typically decreases the residual in $O(N^{-1/2})$ after seeing N samples, the factor of $\frac{1-\gamma}{|\mathcal{S}|}$ earlier would turn into a multiplier of $\frac{|\mathcal{S}|^2}{(1-\gamma)^2}$ in sample complexity. This makes existing saddle-point approaches sample inefficient in comparison with other RL methods like Q-learning [21].

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 [5], 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 γ 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 [5].

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 nonnegative. This term is similar to an equality called the performance difference lemma [32, 33].

Lemma 2. *Let $\mathbf{v}^\pi$ and $\boldsymbol{\mu}^\pi$ denote the value and state-action distribution of some policy π. 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 $\mathbf{d}^{\pi_{\boldsymbol{\mu}}} = (1-\gamma)\mathbf{p} + \gamma \mathbf{P}_{\pi_{\boldsymbol{\mu}}}^\top \mathbf{d}^{\pi_{\boldsymbol{\mu}}}$ from (2), where $\mathbf{P}_{\pi_{\boldsymbol{\mu}}}$ is the transition matrix under $\pi_{\boldsymbol{\mu}}$. But the missing half $\gamma \mathbf{P}_{\pi_{\boldsymbol{\mu}}}^\top \mathbf{d}^{\pi_{\boldsymbol{\mu}}}$ (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 an uniform bound on $\|\mathbf{v}^* - \mathbf{v}^{\pi_{\boldsymbol{\mu}}}\|_\infty$. The counterexample in Proposition 3 was designed to maximize the effect of distribution shift, so that $\boldsymbol{\mu}$ fails to captures state-action pairs with high advantage. To break the curse, we must properly weight the gap between $\mathbf{v}^*$ and $\mathbf{v}^{\pi_{\boldsymbol{\mu}}}$ instead of relying on the uniform bound on $\|\mathbf{v}^* - \mathbf{v}^{\pi_{\boldsymbol{\mu}}}\|_\infty$ as before.

4 THE REDUCTION

The analyses above reveal both good and bad properties of the saddle-point setup in (8). On the one hand, we showed that approximate solutions to the saddle-point problem in (8) can be obtained by running any no-regret algorithm in the single-player COL problem defined in (16); 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 (8) to an approximately optimal policy in RL (Proposition 2), even after imposing strong conditions like (19). At this point, one may wonder if the formulation based on (8) 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 (16) is, in fact, as sample efficient in policy performance as other RL techniques, even without the common constraint in (19) 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* (17)*, which is either the average or the best decision in X_N. Define $y_N^* := (\mathbf{v}^{\hat{\pi}_N}, \boldsymbol{\mu}^*)$. Then $V^{\hat{\pi}_N}(p) \geq V^*(p) - \frac{Regret_N(y_N^*)}{N}$.*

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 $O(\mathrm{Regret}_N(y_N^*)/N)$. Compared with existing results obtained through Proposition 2, the above result removes the factor $(1-\gamma)\min_s p(s)$ and impose *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 $\boldsymbol{\mu}$ 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 [26] or PicCoLO [19]—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 $\boldsymbol{\mu}$ and $\mathbf{v}$. We will present a corollary of Theorem 1, which separates the problem of *learning* $\boldsymbol{\mu}$ 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 $x = (\mathbf{v}, \boldsymbol{\mu}) \in \mathcal{X}$, define $y_x^* := (\mathbf{v}^{\pi_\mu}, \boldsymbol{\mu}^*) \in \mathcal{X}$. 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 (17). 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 [15]: because F is skew-symmetric and $F(y_N^*, \cdot)$ is convex, we have by (18)

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

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

$$\mathcal{X}_\Theta = \{\mathbf{x}_\theta = (\boldsymbol{\phi}_\theta, \boldsymbol{\psi}_\theta) : \boldsymbol{\psi}_\theta \in \mathcal{M}, \theta \in \Theta\}, \qquad (20)$$

where $\boldsymbol{\phi}_\theta$ and $\boldsymbol{\psi}_\theta$ are functions parameterized by $\theta \in \Theta$, and Θ 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 [34–36]. Notice that, in (20), we require only

$\boldsymbol{\psi}_\theta \in \mathcal{M}$, but not $\boldsymbol{\phi}_\theta \in \mathcal{V}$, 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*

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

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_θ, and $\mathrm{Regret}_N(\Theta) := \sum_{n=1}^N l_n(x_n) - \min_{x \in \mathcal{X}_\Theta} \sum_{n=1}^N l_n(x)$.

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*

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

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 θ between $\mathbf{v} = \boldsymbol{\phi}_\theta$ and $\boldsymbol{\mu} = \boldsymbol{\psi}_\theta$, 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 $l_n((\boldsymbol{\phi}_\theta, \boldsymbol{\psi}_\theta))$ remains a convex function in θ, 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 two separate factors: 1) the rate of regret $\mathrm{Regret}_N(\Theta)$ 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 $l_n((\boldsymbol{\phi}_\theta, \boldsymbol{\psi}_\theta))$ as a function of θ, and the choice of Θ needs to ensure (20) 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 (16) 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.

Algorithm 1 Mirror descent for RL

```
Input: $\epsilon$ optimality of the $\gamma$-average return
       $\delta$ maximal failure probability
       generative model of an MDP
Output: $\hat{\pi}_N = \pi^{\hat{\boldsymbol{\mu}}_N}$
$x_1 = (\mathbf{v}_1, \boldsymbol{\mu}_1)$ where $\boldsymbol{\mu}_1$ is uniform and $\mathbf{v}_1 \in \mathcal{V}$
Set $N = \tilde{\Omega}(\frac{|\mathcal{S}||\mathcal{A}| \log(\frac{1}{\delta})}{(1-\gamma)^2 \epsilon^2})$ and $\eta = (1-\gamma)(|\mathcal{S}||\mathcal{A}|N)^{-1/2}$
Set the Bregman divergence as (22)
for $n = 1 \dots N-1$ do
    Sample $g_n$ according to (24)
    Update to $x_{n+1}$ according to (21)
end for
Set $(\hat{\mathbf{v}}_N, \hat{\boldsymbol{\mu}}_N) = \hat{x}_N = \frac{1}{N} \sum_{n=1}^N x_n$
```

Recall that mirror descent is a first-order algorithm, whose update rule can be written as

$$x_{n+1} = \arg\min_{x \in \mathcal{X}} \langle g_n, x \rangle + \tfrac{1}{\eta} B_R(x||x_n) \tag{21}$$

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

$$B_R(x'||x) = \tfrac{1}{2|\mathcal{S}|} \|\mathbf{v}' - \mathbf{v}\|_2^2 + KL(\boldsymbol{\mu}'||\boldsymbol{\mu}) \tag{22}$$

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 (16) and consider a new loss

$$h_n(x) := \mathbf{b}_{\boldsymbol{\mu}_n}^\top \mathbf{v} + \boldsymbol{\mu}^\top (\tfrac{1}{1-\gamma} \mathbf{1} - \mathbf{a}_{\mathbf{v}_n}) \tag{23}$$

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. EXP3 [37]) 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:

$$g_n = \begin{bmatrix} \tilde{\mathbf{p}}_n + \frac{1}{1-\gamma} (\gamma \tilde{\mathbf{P}}_n - \mathbf{E}_n)^\top \tilde{\boldsymbol{\mu}}_n \\ |\mathcal{S}||\mathcal{A}| (\frac{1}{1-\gamma} \hat{\mathbf{1}}_n - \hat{\mathbf{r}}_n - \frac{1}{1-\gamma} (\gamma \hat{\mathbf{P}}_n - \hat{\mathbf{E}}_n) \mathbf{v}_n) \end{bmatrix} \tag{24}$$

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 $\tilde{}$ and $\hat{}$ 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 [3, 5, 10] but with improved performance.

Theorem 2. *With probability $1-\delta$, Algorithm 1 learns an ϵ-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. We provide a sketch here; please refer to Appendix D for details. Let $y_N^* = (\mathbf{v}^{\hat{\pi}_N}, \boldsymbol{\mu}^*)$. We bound the regret in Theorem 1 by the following rearrangement, where the first equality below is because h_n is a constant shift from l_n.

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

We recognize the first term is a martingale, because x_n does not depend on g_n. Therefore, we can appeal to a Bernstein-type martingale concentration and prove it is in $\tilde{O}(\frac{\sqrt{N|\mathcal{S}||\mathcal{A}|\log(\frac{1}{\delta})}}{1-\gamma})$. For the second term, by treating $g_n^\top x$ as the per-round loss, we can use standard regret analysis of mirror descent and show a bound in $\tilde{O}(\frac{\sqrt{N|\mathcal{S}||\mathcal{A}|}}{1-\gamma})$. For the third term, because $\mathbf{v}^{\hat{\pi}_N}$ in $y_N^* = (\mathbf{v}^{\hat{\pi}_N}, \boldsymbol{\mu}^*)$ depends on $\{g_n\}_{n=1}^N$, it is *not* a martingale. Nonetheless, we are able to handle it through a union bound and show it is again no more than $\tilde{O}(\frac{\sqrt{N|\mathcal{S}||\mathcal{A}|\log(\frac{1}{\delta})}}{1-\gamma})$. Despite the union bound, it 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 high-probability regret bound into Theorem 1 to obtain the desired claim.

5.2 Extension to Function Approximators

The above algorithm assumes the 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 $(\boldsymbol{\theta}_v, \boldsymbol{\theta}_\mu) \in \Theta$.

Overall the algorithm stays the same, except the gradient is computed by chain-rule, which can be done in $O(dim(\Theta))$ time and space. While this seems worse, the computational complexity per update actually improves to $O(dim(\Theta))$ from the slow $O(|\mathcal{S}||\mathcal{A}|)$ (required before for the projection in (21)), as now we only optimize in Θ. Moreover, we prove that its sample complexity is also 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 Θ 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 proof is detailed in Appendix E, which mainly follows Section 5.1. First, we choose some Θ to satisfy (20) so we can use Corollary 1 to reduce the problem into regret minimization. To make the sample complexity independent of $|\mathcal{S}|$,$|\mathcal{A}|$, the key is to uniformly sample over the columns of $\boldsymbol{\Psi}$ (instead of over all states and actions like (24)) when computing unbiased estimates of $\nabla_{\boldsymbol{\theta}_\mu} h_n((\boldsymbol{\theta}_v, \boldsymbol{\theta}_\mu))$. The intuition is that we should only focus on the places our basis functions care about (of size $dim(\Theta)$), instead of wasting efforts to visit all possible combinations (of size $|\mathcal{S}||\mathcal{A}|$).

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 evidences the strength of the online learning perspective. As a future work, we believe even faster learning in RL is possible by leveraging control variate for variance reduction and by applying more advanced online techniques [18, 19] that exploit the continuity in COL to predict future gradients.

Acknowledgements

This research is partially supported by NVIDIA Graduate Fellowship.

References

[1] Alan S Manne et al. Linear programming and sequential decision models. Technical report, Cowles Foundation for Research in Economics, Yale University, 1959.

[2] Eric V Denardo and Bennett L Fox. Multichain Markov renewal programs. *SIAM Journal on Applied Mathematics*, 16(3):468–487, 1968.

[3] 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.

[4] Yichen Chen and Mengdi Wang. Stochastic primal-dual methods and sample complexity of reinforcement learning. *arXiv preprint arXiv:1612.02516*, 2016.

[5] Mengdi Wang. Randomized linear programming solves the discounted Markov decision problem in nearly-linear running time. *ArXiv e-prints*, 2017.

[6] Donghwan Lee and Niao He. Stochastic primal-dual Q-learning. *arXiv preprint arXiv:1810.08298*, 2018.

[7] Mengdi Wang. Primal-dual π learning: Sample complexity and sublinear run time for ergodic Markov decision problems. *arXiv preprint arXiv:1710.06100*, 2017.

[8] 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.

[9] Bo Dai, Albert Shaw, Niao He, Lihong Li, and Le Song. Boosting the actor with dual critic. In *International Conference on Learning Representation*, 2018.

[10] Yichen Chen, Lihong Li, and Mengdi Wang. Scalable bilinear π learning using state and action features. *arXiv preprint arXiv:1804.10328*, 2018.

[11] 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.

[12] Jyrki Kivinen and Manfred K Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *information and computation*, 132(1): 1–63, 1997.

[13] Geoffrey J Gordon. Regret bounds for prediction problems. In *Conference on Learning Theory*, volume 99, pages 29–40, 1999.

[14] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning*, pages 928–936, 2003.

[15] Ching-An Cheng, Jonathan Lee, Ken Goldberg, and Byron Boots. Online learning with continuous variations: Dynamic regret and reductions. *arXiv preprint arXiv:1902.07286*, 2019.

[16] Eugen Blum. From optimization and variational inequalities to equilibrium problems. *Math. student*, 63:123–145, 1994.

[17] M Bianchi and S Schaible. Generalized monotone bifunctions and equilibrium problems. *Journal of Optimization Theory and Applications*, 90(1): 31–43, 1996.

[18] Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. *arXiv preprint arXiv:1208.3728*, 2012.

[19] Ching-An Cheng, Xinyan Yan, Nathan Ratliff, and Byron Boots. Predictor-corrector policy optimization. In *International Conference on Machine Learning*, 2019.

[20] 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.

[21] Sham Machandranath Kakade et al. *On the sample complexity of reinforcement learning*. PhD thesis, University of London London, England, 2003.

[22] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[23] Onésimo Hernández-Lerma and Jean B Lasserre. *Discrete-time Markov control processes: basic optimality criteria*, volume 30. Springer Science & Business Media, 2012.

[24] Richard Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–515, 1954.

[25] 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.

[26] Anatoli Juditsky, Arkadi Nemirovski, and Claire Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1):17–58, 2011.

[27] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[28] Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[29] Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[30] Alejandro Jofré and Roger J-B Wets. Variational convergence of bifunctions: motivating applications. *SIAM Journal on Optimization*, 24(4):1952–1979, 2014.

[31] 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.

[32] 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.

[33] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, volume 2, pages 267–274, 2002.

[34] 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.

[35] Ching-An Cheng and Byron Boots. Convergence of value aggregation for imitation learning. *International Conference on Artificial Intelligence and Statistics*, 2018.

[36] Ching-An Cheng, Xinyan Yan, Evangelos A Theodorou, and Byron Boots. Accelerating imitation learning with predictive models. In *International Conference onArtificial Intelligence and Statistics*, 2019.

[37] 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`.

[38] Colin McDiarmid. Concentration. In *Probabilistic methods for algorithmic discrete mathematics*, pages 195–248. Springer, 1998.