Faster Rates, Adaptive Algorithms, and Finite-Time Bounds for Linear Composition Optimization and Gradient TD Learning

Anant Raj[1] **Pooria Joulani**[2] **András György**[2] **Csaba Szepesvári**[2,3]
[1]INRIA, Paris, France [2]DeepMind, London, UK [3]Univ. of Alberta, Edmonton, Canada

Abstract

Gradient temporal difference (GTD) algorithms are provably convergent policy evaluation methods for off-policy reinforcement learning. Despite much progress, proper tuning of the stochastic approximation methods used to solve the resulting saddle point optimization problem requires the knowledge of several (unknown) problem-dependent parameters. In this paper we apply adaptive step-size tuning strategies to greatly reduce this dependence on prior knowledge, and provide algorithms with adaptive convergence guarantees. In addition, we use the underlying refined analysis technique to obtain new $\mathcal{O}(1/T)$ rates that do not depend on the strong-convexity parameter of the problem, and also apply to the Markov noise setting, as well as the unbounded i.i.d. noise setting.

1 INTRODUCTION

Gradient temporal difference (GTD) algorithms (Maei, 2011) are policy evaluation methods for reinforcement learning (RL). Unlike the traditionally successful "semi-gradient" TD methods (Sutton, 1988), GTD algorithms are provably convergent even in the so-called off-policy learning setting (Sutton et al., 2008, 2009; Maei, 2011). The three basic linear GTD algorithms, known, respectively, as (linear) GTD, GTD2 and TDC, can be viewed and analyzed as stochastic approximation methods applied to the following optimization problem:

$$\min_{\theta\in\Theta} \ell(\theta)\,, \quad \text{where } \ell(\theta) := \frac{1}{2}\left\|b - A\theta\right\|^2_{M^{-1}}\,, \qquad (1)$$

where θ denotes the parameters of the linearly-represented value function being learned, Θ is a convex set (equal to $\mathbb{R}^d$ in the unconstrained case), $b \in \mathbb{R}^d$, $A \in \mathbb{R}^{d\times d}$ and $M \in \mathbb{R}^{d\times d}$ are quantities determined by the policy-evaluation problem at hand (cf. Section 7), and $\|y\|^2_C := y^\top C y$ for any $C \in \mathbb{R}^{d\times d}$ and $y \in \mathbb{R}^d$.

In this paper, we study iterative algorithms for solving problem (1) in its generic form.[1] In particular, we study algorithms that, at iterations $t = 1, 2, \ldots, T$, can only access estimates[2] b_t, A_t and M_t, respectively, of b, A, and M. Under this information structure, problem (1) is a special case of *composition optimization* (Wang et al., 2017a). While in standard first-order optimization an algorithm (e.g., stochastic gradient descent) observes estimates of $\nabla\ell(\theta) = A^\top M^{-1}(A\theta - b)$, our situation is more complicated: due to the presence of $A^\top M^{-1}$ in the expression for the gradient, plugging in even independent unbiased estimates of the parameters b, A, and M does not lead to an unbiased estimate of $\nabla\ell(\theta)$. As such, under this information structure, iterative algorithms for problem 1 (including GTD methods) usually consist of two iterative updates: in addition to a parameter estimate θ_t, an auxiliary variable y_t is also maintained, usually for estimating the factor $M^{-1}(b - A\theta)$ in $\nabla\ell(\theta)$.

Our goal is to answer two questions which, despite much progress, remain unanswered by previous work:

1. **Prior knowledge about b, A, and M.** Can we guarantee convergence for algorithms with less dependence on prior knowledge? As detailed in Table 1, existing analyses of linear GTD / GTD2 / TDC prove convergence only if the step sizes of these methods lie in a restricted interval that depends on prior knowledge of upper-bounds λ_A and λ_M (holding uniformly over t) on the eigenvalues of $A_t A_t^\top$ and M_t, as well as lower-bounds β and μ on the eigenvalues, respectively, of M and $A^\top M^{-1} A$, among other parameters. In contrast, standard adaptive stochastic optimization methods, such as AdaGrad (Duchi et al., 2011; McMahan and Streeter, 2010), allow one

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

[1]See Section 2 for the formal conditions and the assumptions.

[2]Note that the estimates may not be i.i.d. over time.

to reduce the dependence of the step sizes on such prior knowledge, and provide adaptive convergence guarantees (rather than a worst-case upper-bound) that depend on the actual problem at hand. Such adaptation to unknown problem parameters are not supported by the existing analyses (see Section 1.2).

2. **Dependence on problem parameters.** Can we provide guarantees on the convergence of the function value $\ell(\bar{\theta}_T)$ (where $\bar{\theta}_T$ is the output of the algorithm after T iterations), or the iterates $\|\theta_t - \theta^*\|^2$, that scale with β and μ in the same way as in first-order optimization? Currently, the final convergence guarantee provided by previous work depends in an obscure, sub-optimal way on β and μ (even when the aforementioned prior knowledge is provided). In comparison, when (as in our case) the Hessian of $\ell(\theta)$ has eigenvalues lower-bounded by μ, stochastic gradient descent guarantees a rate of $\ell(\bar{\theta}_T) \leq \mathcal{O}(1/(\mu T))$, and for quadratic objectives, Bach and Moulines (2013) show a rate of $\mathcal{O}(1/T)$ independently of μ. Existing analyses do not provide such a rate for the quadratic objective of problem (1).

In this paper, we provide an affirmative answer to these questions.

1.1 Contributions

Using a range of new step-size schedules and algorithms, and a refinement of the analysis technique of Liu et al. (2018):

- We present adaptive step-size tuning strategies that require much less prior knowledge than previous work. Given only knowledge of λ_M, these algorithms guarantee $\ell(\bar{\theta}_T)$ to converge at a rate of $\mathcal{O}(1/\sqrt{T})$ (Theorem 2), with an improved rate of $\mathcal{O}(\frac{1}{\beta T})$ given further knowledge of β (Theorem 3). Crucially, in the context of (off-policy) policy evaluation, these rates are achieved with step sizes independent of the maximum policy ratio $\rho_{\max}$, with the $\mathcal{O}(1/\sqrt{T})$ rate requiring only a bound on the feature norm $\|\phi\|$ of the RL problem; see Section 7.

- With access to similar prior knowledge as previous work (i.e., β, μ, λ_M, and λ_A^2/β), we present a restarting scheme that guarantees $\ell(\bar{\theta}_T)$ to converge at a rate of $\mathcal{O}(\frac{1}{\beta T})$[3] when the samples b_t, A_t, and M_t are obtained under either i.i.d. (Theorem 1) or Markov noise (Theorems 4 and 5). This puts our convergence guarantee in between the standard $\frac{1}{\mu T}$ rate of SGD for strongly-convex functions, and the special-case rate $\mathcal{O}(\frac{1}{T})$ of SGD for least-square objectives (Bach and Moulines, 2013).[4]

On the technical side, our results rely on a refined analysis framework for primal-dual algorithms, which may be of independent interest. We provide a tight decomposition of the optimization gap (instead of the min-max saddle-point gap considered by Liu et al., 2015, 2018), which features extra flexibility to exploit the curvature of the problem and to use the recent advances from the online linear optimization (OLO) literature.

In the sequel, we also relax previous requirements of projecting the dual variables y_t or, alternatively, specify conditions required for the dual constraint set $\mathcal{Y}$ to ensure convergence to the solution of problem (1).

1.2 Related Work

In recent years, there has been a new interest in the analysis of algorithms for problem (1) with application to policy-evaluation. Due to the numerous prior work, we delegate the complete study of related work to the extended version of this paper (Raj et al., 2022), and in this section provide a general overview of the state of the art with specific examples.

Previous work that prove finite-time (i.e., non-asymptotic) convergence bounds for problem (1) can be broadly categorized based on the quantity they control:

- **Bounding the iterates** $\|\theta_t - \theta^*\|$: This group of papers (e.g., Dalal et al., 2017, 2018, 2020; Gupta et al., 2019; Kaledin et al., 2020) study the iterative updates for θ_t and y_t as a (two-time-scale) update. Their bound can be turned into a bound on the objective by multiplying it by λ_A^2/β, since $\ell(\theta) - \ell(\theta^*) \leq \frac{\lambda_A^2}{2\beta}\|\theta - \theta^*\|^2$. These results typically hold only for a limited range of (non-adaptive) step sizes, but work for unbounded constraint sets (with the exception of Dalal et al., 2018; Xu et al., 2019).

- **Bounding the objective** $\ell(\bar{\theta}_T) - \ell(\theta^*)$: Starting with the seminal works of Liu et al. (2015, 2018), this group of papers (e.g., Du et al., 2017; Peng et al., 2019) reduce problem (1) to a saddle-point problem (see Section 2), and study the convergence of stochastic gradient descent-ascent (SGDA) on this

[3] This is a stronger result than the $\mathcal{O}(1/T)$ rates by Du et al. (2017); Peng et al. (2019) that are based on combining GTD with SVRG (Johnson and Zhang, 2013): The SVRG-based methods are applicable only to the finite-batch learning setting, and need to frequently access *all* data points in the fixed batch to compute a full gradient. In contrast, our method applies to infinite online sequences with Markov noise.

[4] Recall that SGD has direct sampling of the gradient of $\ell(\theta)$ (which is not possible in problem 1 with access only to estimates A_t, b_t and M_t).

saddle-point problem. Typically, to be able to use off-the-shelf optimization guarantees, these papers consider only the case when the constraint set Θ is bounded. For the same reason, these papers typically also project the dual variable y_t to a compact set $\mathcal{Y}$. As a result of the saddle-point formulation, these papers also typically lose the benefits of the curvature of the problem, hence obtaining only $\mathcal{O}(1/\sqrt{T})$ rates. Notably, for the more general setting of composition optimization, Wang et al. (2017b) obtain an $\mathcal{O}(1/T)$ rate for the objective assuming that the gradients have bounded norms, which in general is not possible in the GTD set-up unless the iterates are enforced to remain in a bounded set. In addition, the constant in their convergence rate seemingly scales with $\Omega(1/\mu)$. Note that convergence rates on the objective can, in general, be turned[5] into convergence rates on $\|\bar{\theta}_T - \theta^*\|^2$, since $\ell(\theta) - \ell(\theta^*) \geq \frac{\mu}{2}\|\theta - \theta^*\|^2$.

In the present paper, we extend the second group of analyses to work without projection of θ_t and/or y_t and to obtain $\mathcal{O}(1/T)$ rates, while deriving the exact dependence of the rate on the problem parameters and providing adaptive step-size schedules to relax the prior-knowledge requirements of the first group of papers. Table 1 compares our results with the most relevant previous works; further discussion of the related literature, including recent works by Wang et al. (2018); Hu and Syed (2019); Huang and Zhang (2021); Kotsalis et al. (2020a,b); Doan (2021) is presented in the extended version of this paper (Raj et al., 2022).

2 PRELIMINARIES

Notation. The set $\{1, 2, \ldots, n\}$ is denoted by $[n]$. We denote the Euclidean norm on $\mathbb{R}^d$ by $\|\cdot\|$, and for any matrix $C \in \mathbb{R}^{d\times d}$, we denote $\|C\| = \|C\|_2 = \max_{x\in\mathbb{R}^d\setminus\{0\}} \frac{\|Cx\|}{\|x\|}$ and $\|y\|_C = y^\top C y$ for any $y \in \mathbb{R}^d$. The largest (respectively, smallest) singular value of a matrix C is denoted by $\sigma_{\max}(C)$ (respectively, $\sigma_{\min}(C)$). For $\mu > 0$, a differentiable function f is μ-strongly-convex if $f(x) - f(y) - \langle \nabla f(y), x - y\rangle \geq \frac{\mu}{2}\|x - y\|^2$. For $\beta > 0$, a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is β-strongly-concave if $-f$ is β-strongly-convex. We use $\mathcal{F}_t = \sigma\left(\{b_s, A_s, M_s\}_{s=1}^{t-1}\right)$ to denote the sigma-field of all observations prior to time t, and $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot|\mathcal{F}_t]$ to denote conditional expectation given $\mathcal{F}_t$. For a closed convex set $\mathcal{X} \subset \mathbb{R}^d$ and $x \in \mathcal{X}$, $\mathcal{P}_{\mathcal{X}}(x) = \arg\min_{y\in\mathcal{X}} \|y - x\|$ denotes the Euclidean projection of x to $\mathcal{X}$ (which is unique).

[5]However, obtaining bounds on $\|\theta_t - \theta^*\|$ (rather than $\|\bar{\theta}_T - \theta^*\|$) may require further techniques, see, e.g., Shamir and Zhang (2013).

Assumptions. Throughout this paper, we make the following assumptions:

Assumption 2.1. M is symmetric and positive-definite, A is invertible, and there exist[6] $\mu, \beta, \lambda_A, \lambda_M > 0$ and $B \geq 0$ such that

1. $\mu \leq \sigma_{\min}\left(A^\top M^{-1} A\right)$ and $\beta \leq \sigma_{\min}(M)$;
2. $\sigma_{\max}(A_t) \leq \lambda_A$ and $\sigma_{\max}(M_t) \leq \lambda_M$ for all $t \in [T]$;
3. $\|b_t\| \leq B$ for all $t \in [T]$.

Optimal primal and dual variables. Let $\theta^\star$ denote the minimizer of $\ell(\theta)$ in problem 1, which exists and is unique by Assumption 2.1 (as it implies that ℓ is μ-strongly-convex). Following Liu et al. (2018), we can express ℓ as $\ell(\theta) = \arg\max_{y\in\mathbb{R}^d} L(\theta, y)$, where

$$L(\theta, y) = \langle b - A\theta, y\rangle - \frac{1}{2}\|y\|_M^2 .$$

For $\theta \in \mathbb{R}^d$, we denote $y^\star(\theta) = \arg\max_{y\in\mathbb{R}^d} L(\theta, y)$, which is well-defined under Assumption 2.1 (since it implies that $L(\theta, \cdot)$ is β-strongly-concave as M is assumed to be positive-definite). We define $y^\star = y^\star(\theta^\star)$.

We consider two settings regarding the noise in the estimates b_t, A_t and M_t:

The i.i.d. noise setting. In this setting, we assume that the estimates b_t, A_t, M_t are conditionally unbiased given the past observations:

Assumption 2.2. For all $t \in [T]$,

$$\mathbb{E}\left[A_t|\mathcal{F}_t\right] = A,\ \mathbb{E}\left[M_t|\mathcal{F}_t\right] = M,\ \text{and } \mathbb{E}\left[b_t|\mathcal{F}_t\right] = b\,.$$

By Jensen's inequality and the convexity of $\|\cdot\|^2$, Assumption 2.1 and Assumption 2.2 imply $\|A\|^2 \leq \lambda_A^2$, $\|M\|^2 \leq \lambda_M^2$, and $\|b\|^2 \leq B^2$.

The Markov setting. In the Markov noise setting, we replace Assumption 2.2 with a weaker assumption on the underlying stochastic process. With a slight abuse of notation, let $A : \Xi \to \mathbb{R}^{d\times d}, M : \Xi \to \mathbb{R}^{d\times d}$, and $b : \Xi \to \mathbb{R}^d$ be random variables defined on the state space Ξ of a Markov process. We denote the state of this Markov process at time t by ξ_t, so that $A_t = A(\xi_t)$, $b_t = b(\xi_t)$, and $M_t = M(\xi_t)$. Our main assumption, made by all previous work we are aware of, is that the stochastic process $\{\xi_t\}$ has a finite mixing-time:

Assumption 2.3 (Mixing time). There exists a distribution P_∞ over Ξ (called the steady-state distribution

[6]Note that any of these problem parameters may be unknown to the algorithm.

	Prior info	Rate	Noise	Projected	Notes
Gupta et al. (2019)	$\beta, \mu, \lambda_A, \lambda_M$	$1/(\mu T)$	Markov	No	No adaptive guarantees
Xu et al. (2019)	$\beta, \mu, \|\phi\|, (\lambda_A/\sigma_{\min}(A), \|b\|)$	$1/(\mu T^{1-\delta})$	Markov	Yes	Normalizes the features ϕ
Dalal et al. (2018)	$\beta, \mu, \lambda_A, \lambda_M, \|b\|$	$1/T^{1/3-\delta}$	Markov	Yes	$\log(T)$ many projections
Kaledin et al. (2020)	$\beta, \mu, \lambda_A, \lambda_M, \|b\|$	$\Omega\left(\frac{d}{\min\{\mu,\beta\}\mu^2}\right)/T$	Markov	No	Large constants in the rate
Theorem 1	$\lambda_A^2/\beta, \lambda_M, \beta, \mu$	$1/(\beta T)$	i.i.d	No	Independent of μ
Theorem 2	λ_A^2/β^2 OR λ_M	$1/\sqrt{T}$	i.i.d	Yes	Adaptive step size
Theorem 3	$(\lambda_A^2/\beta$ OR $\lambda_M)$, β	$1/(\beta T)$	i.i.d	Yes	Adaptive step size
Theorem 4	λ_A^2/β, β	$1/(\beta T)$	Markov	Yes	Independent of μ
Theorem 5	λ_A^2/β, λ_M, β, μ	$1/(\beta T)$	Markov	No	Independent of μ

Table 1: Comparison with previous work. λ_A and λ_M denote upper-bounds (holding uniformly over t) on the eigenvalues of $A_t A_t^\top$ and M_t, while β and μ are lower-bounds on the eigenvalues, respectively, of M and $A^\top M^{-1} A$. The rates above the double line correspond to the error in the iterate, while our results, below, the double line, to the error in the objective. The former imply a bound on the latter with an extra factor of λ_A^2/β, while the latter imply a bound on the former with an extra $1/\mu$ factor (see the discussion in Section 1.2). **Gupta et al. (2019)** assume $\|A_t\|, \|M_t\|$, and $\|A^\top M^{-1} A\|$ are upper-bounded by 1, hence need $\lambda_M^2 \vee \lambda_A^2 \vee \lambda_A^2/\beta$ for re-scaling; their (constant) step size and finite-time bound further depend on $1/(\mu \wedge \beta)$ (λ_{max} in their notation). **Xu et al. (2019)** need a bound on $\|\phi(s)\|_2$ for normalizing the features, and β, μ for their step sizes. In addition, to convert their results to the unconstrained setting, they need to know $\|b\|$ and $\sigma_{\min}(A)$ or λ_A/β for setting the projection radii. **Dalal et al. (2018)** provide a guarantee that holds only after a large-enough number of iterations depending on the problem parameters. Also, the $\mathcal{O}(1/T)$ bound of **Kaledin et al. (2020)** depends on the problem parameters in obscure ways, but the constant is at least $\Omega\left(\frac{d}{\min\{\mu,\beta\}\mu^2}\right)$. In comparison, **Theorem 1**, using the same prior knowledge with i.i.d. samples, achieves an $\mathcal{O}(1/T)$ rate independent of μ (to our knowledge the first such rate for GTD). **Theorems 2 and 3** trade-off the knowledge of different hyper-parameters when a bounded constraint set Θ is given. Similarly to Xu et al. (2019), the corresponding algorithms can solve the unconstrained version of problem (1) given extra prior knowledge of $\|b\|$ and $\sigma_{\min}(A)$ or $\lambda_A/(\beta\mu)$. **Theorem 4 and 5** for the Markov noise setting are the only ones we are aware of that obtain a μ-independent $1/T$ rate with this level of prior knowledge.

of the Markov process) such that for all $\Delta > 0$, with probability 1,

$$\tau_0(\Delta) := \sup_{t \geq 1} \inf\{\tau > 0 : d_{TV}(P_{[t]}^{t+\tau}, P_\infty) \leq \Delta\} < \infty\,,$$

where $P_{[t]}^s$ denotes the conditional distribution of ξ_s given $\mathcal{F}_t$, and d_{TV} denotes the total variation distance.

2.1 Online Linear Optimization

Online linear optimization (OLO) provides a generic, flexible framework for analyzing the performance of the iterative update rules used in optimization algorithms. An OLO algorithm consists of an update rule for any time step $t \in [T]$, mapping the previously received *linear* losses, given by vectors $g_s, s \in [t]$, to the next iterate x_{t+1}. Perhaps the simplest example of such an update rule is (projected) online gradient descent (OGD), (see, e.g., Zinkevich, 2003):

$$x_{t+1} = \mathcal{P}_\mathcal{X}\left(x_t - \frac{1}{\eta_t} g_t\right), \qquad t \in [T]\,, \tag{2}$$

or its alternative variant known as Dual Averaging (DA):[7]

$$x_{t+1} = \mathcal{P}_\mathcal{X}\left(x_1 - \frac{1}{\eta_t} g_{1:t}\right), \qquad t \in [T]\,, \tag{3}$$

where $\mathcal{X}$ is a closed convex set and $x_1 \in \mathcal{X}$ is an arbitrary initial point. The *step size* $1/\eta_t$ controls how fast the algorithm changes its iterates, and tuning it adaptively using the observed $g_s, s \in [t]$, with

$$\eta_t = \sqrt{\eta_0^2 + \eta^2 \sum_{s=1}^{t} \|g_s\|^2} \tag{4}$$

for some $\eta, \eta_0 \geq 0$, results in a variant of the AdaGrad update (Duchi et al., 2011; McMahan and Streeter, 2010).

OLO is concerned with the cumulative excess (linear) loss of these iterates compared to a fixed iterate $x^* \in \mathcal{X}$. This loss difference is referred to as the *regret* of the

[7] Generalized versions of the OGD update rule are commonly referred to as mirror descent, while generalized DA updates are also known as follow the regularized leader (FTRL) or lazy-projection mirror descent (Hazan, 2019).

OLO algorithm, and is defined formally as

$$R_T^x(x^*) = \sum_{t=1}^{T} \langle g_t, x_t - x^* \rangle \ .$$

In particular, it has been shown (see, e.g., Hazan, 2019; Orabona, 2019; Joulani et al., 2020) that the OGD update enjoys the following regret bound for any $T \geq 1$ and non-increasing step sizes, i.e., for $\eta_t \geq \eta_{t-1}$ for all $t \in [T]$ with $\eta_0 = 0$:

$$\begin{aligned} R_T^x(x^*) &\leq \sum_{t=1}^{T} \left(\frac{\eta_t - \eta_{t-1}}{2} \|x_t - x^*\|^2 + \frac{1}{2\eta_t} \|g_t\|^2 \right) \\ &\leq \frac{\eta_T}{2} r_x^2 + \sum_{t=1}^{T} \frac{1}{2\eta_t} \|g_t\|^2 \,, \end{aligned}$$

where $r_x = \sup_{x_1, x_2 \in \mathcal{X}} \|x_1 - x_2\|$, and a data-dependent regret bound of

$$R_T^x(x^*) \leq \left(\frac{\eta}{2} r_x^2 + \frac{1}{\eta} \right) \sqrt{\sum_{t=1}^{T} \|g_t\|^2} \,,$$

using the AdaGrad step sizes (4) with $\eta_0 = 0$. Similarly, for any $T \geq 1$, the DA update ensures

$$R_T^x(x^*) \leq \frac{\eta_{T-1}}{2} \|x^*\|^2 + \sum_{t=1}^{T} \frac{1}{2\eta_{t-1}} \|g_t\|^2 \,,$$

leading to a data-dependent regret bound of

$$\begin{aligned} R_T^x(x^*) \leq \frac{\eta_0}{2} \|x^*\|^2 + \max_{t \in [T]} \frac{\|g_t\|^2}{2\eta_0} \\ + \left(\frac{\eta}{2} \|x^*\|^2 + \frac{1}{\eta} \right) \sqrt{\sum_{t=1}^{T} \|g_t\|^2} \,, \end{aligned}$$

when using AdaGrad step sizes (4) with $\eta_0 > 0$.

The regret view is useful because it abstracts away the origins of g_t, and views it simply as the input to the algorithm. The only thing we need to do in an application, such as online convex optimization (Zinkevich, 2003; Hazan, 2019), stochastic optimization (Hazan, 2019; Joulani et al., 2020), or saddle-point computation (Juditsky et al., 2011), is to relate our performance metric of choice for the given application to the regret of the update rule in terms of the information g_t that is fed into it. In the next section, we show a refined reduction of this nature that allows us to exploit the structure of the saddle-point problem arising from a primal-dual reformulation of the optimization objective (through L).

3 FROM SADDLE POINTS TO ONLINE LINEAR OPTIMIZATION

Our analysis builds on a refined decomposition of the *optimization error* $\ell(\bar{\theta}) - \ell(\theta^*)$, where $\bar{\theta}$ denotes the average of the iterates produced by the primal algorithm. The new decomposition isolates (but retains) the effect of the curvature of the problem, as captured by the matrices M and A, from the individual performances of the underlying OLO algorithms that receive (possibly biased) estimates of the gradients of L and come up with θ_t and y_t in the primal-dual setup.

Lemma 3.1 (Error decomposition). Consider arbitrary sequences of points $\theta_t \in \Theta$, $y_t \in \mathcal{Y}$, $z_t \in \mathcal{Y}$, $g_t^\theta \in \mathbb{R}^d$, and $g_t^y \in \mathbb{R}^d$, $t \in [T]$. Let $\bar{\theta} = \frac{1}{T} \sum_{t=1}^{T} \theta_t$ be the average of the points θ_t, let $\bar{y}^* = \arg\max_{y \in \mathbb{R}^d} L\left(\bar{\theta}, y\right)$, and suppose that $\mathcal{Y}$ is such that $\bar{y}^* \in \mathcal{Y}$. For $t \in [T]$, let $B_t = \frac{1}{2}\|y_t\|_M^2$ and $\bar{B}_t = \frac{1}{2}\|y_t - \bar{y}^*\|_M^2$, and define $\sigma_t^\theta = \nabla_\theta L(\theta_t, y_t) - g_t^\theta$ and $\sigma_t^y = \nabla_y L(\theta_t, y_t) - g_t^y$. Then,

$$\ell(\bar{\theta}) - \ell(\theta^*) = \frac{\epsilon_{1:T}}{T} + \frac{R_T^\theta + R_T^y + R_T^\sigma}{T} - \frac{B_{1:T} - \bar{B}_{1:T}}{T} \,, \tag{5}$$

where $\epsilon_t = \left\langle \sigma_t^\theta, \theta_t - \theta^* \right\rangle + \langle \sigma_t^y, z_t - y_t \rangle$, $t \in [T]$, while $R_T^\theta = \sum_{t=1}^{T} \left\langle g_t^\theta, \theta_t - \theta^* \right\rangle$, $R_T^y = \sum_{t=1}^{T} \langle -g_t^y, y_t - \bar{y}^* \rangle$, and $R_T^\sigma = \sum_{t=1}^{T} \langle -\sigma_t^y, z_t - \bar{y}^* \rangle$.

Remark. (i) In the decomposition above, g_t^θ and g_t^y serve, respectively, as the estimates of the gradients of L at (θ_t, y_t) w.r.t. θ and y, respectively. In particular, using the estimates b_t, A_t and M_t, we have

$$g_t^y = b_t - A_t \theta_t - M_t y_t,$$

and

$$g_t^\theta = -{A_t}^\top y_t \,.$$

The terms ϵ_t then capture the noise in this estimate, so that $\mathbb{E}\{\epsilon_t\}$ is zero when the noise at time t is independent of the history up to time t (the i.i.d. setting), but possibly non-zero (and controlled via a separate argument) in the Markov noise setting. The terms R_T^θ and R_T^y are the regrets of the OLO algorithms producing θ_t and y_t when sequentially fed with the linear feedback g_t^θ and g_t^y, respectively. Following Juditsky et al. (2011), the term R_T^σ is the regret of an *imaginary* OLO algorithm which is sequentially fed with σ_t^y and produces iterates z_t.; this ensures ϵ_t does not involve $\bar{y}^*$, which depends on the whole history and complicates the analysis of noise.

(ii) Compared to previous work, the decomposition above applies directly to the optimization error of the primal-dual scheme, as opposed to the variational-inequality error pertaining to the prima-dual gradient

operator. This enables us to take advantage of the curvature of the problem through the negative terms $-B_t$ and $-\bar{B}_t$; this is instrumental for reducing the contributions of the regret terms to the final error bound, and obtaining not only the $\mathcal{O}(1/T)$ convergence guarantees but also simpler step-size configurations and smaller final convergence bounds.

The analysis then works as follows: we control the terms ϵ_t using the i.i.d. or mixing properties of the estimates and the underlying stochastic process $\{\xi_t\}$. Furthermore, we upper-bound the regret terms in (5) using the OLO regret bounds of the corresponding updates for θ_t and y_t (Section 2.1). This leaves us with the terms arising from those regret bounds, which involve the iterates as well as the gradients $\|g_t^\theta\|^2$ and $\|g_t^y\|^2$. We then control these terms using the generic bounds presented below, possibly using the matching negative terms $-B_t$ and $-\bar{B}_t$ from (5) to further reduce their effect in the final bound on $\ell(\bar{\theta}) - \ell(\theta^*)$.

Generic bounds on the regret terms. Next, we upper-bound the terms arising from the OLO regret guarantees of Section 2.1.

Lemma 3.2 (Bound on $\|g_t^\theta\|^2$). Recall $B_t = \frac{1}{2}\|y_t\|_M^2$. For any $t \in [T]$, we have $\|g_t^\theta\|^2 \leq \frac{2\lambda_A^2}{\beta} B_t$.

Lemma 3.3 (Bound on $\|g_t^y\|^2$). Assume the i.i.d. setting, and let σ_*^2 be an upper-bound on $\mathbb{E}_t[\|b_t - A_t\theta^*\|^2]$. Then, for any $t \in [T]$,

$$\mathbb{E}_t\left[\|g_t^y\|^2\right] \leq 3\left(\sigma_*^2 + \lambda_A^2\|\theta_t - \theta^*\|^2 + 2\lambda_M B_t\right) .$$

In addition, if θ_t is given by DA or OGD with the step-size sequence η_t^θ, then for any $t \in [T]$,

$$\mathbb{E}_t\left[\|g_t^y\|^2\right] \leq 4\left(\sigma_*^2 + \lambda_A^2\|\theta_1 - \theta^*\|^2 + 2\lambda_M B_t + \frac{2\lambda_A^4 \sum_{s=1}^{t-1}(t-1)B_s}{\beta\left(\eta_t^\theta\right)^2}\right) .$$

Lemma 3.4 (Bound on $\|\bar{y}_T^*\|^2$). We have $\|\bar{y}_T^*\|^2 \leq \frac{2}{\beta}\left(\ell(\bar{\theta}_T) - \ell(\theta^*)\right)$.

Lemma 3.5 (Bound on $\|\theta - \theta^*\|^2$). For any $\theta \in \mathbb{R}^d$, we have $\|\theta - \theta^*\|^2 \leq \frac{2}{\mu}\left(\ell(\theta) - \ell(\theta^*)\right)$.

Equipped with the generic reduction to OLO and the above bounds on the terms that appear in the corresponding OLO regret bounds, we are ready to mix and match OLO algorithms and tuning strategies to obtain a range of new convergence results for problem (1).

4 CONVERGENCE RESULTS WITH UNBOUNDED Θ IN THE I.I.D. SETTING

We start our results by showing finite time bounds for the case of unbounded Θ in the i.i.d. setting.

In this section, we provide a new primal-dual algorithm that converges at a rate of $\mathcal{O}(1/T)$, rather than the usual rate of $\mathcal{O}(1/\sqrt{T})$, requires less prior knowledge, and results in better dependence on the parameters than earlier methods with $\mathcal{O}(1/T)$ convergence guarantees. The two main features of our algorithm, namely restarting (i.e., the use of epochs) and unprojected updates, have appeared in previous work. However, we can now show that this simple update strategy, given in Algorithm 1, enjoys a bound of order $\mathcal{O}(1/(\beta T))$ on the objective $\ell(\theta)$, unlike previous work which obtain a bound (on the iterate norms r.t. the objective) that typically scales with $1/\mu$.

Restarting. The algorithm works in S epochs, where epoch s, $s = 1, 2, \ldots, S$, consists of T_s rounds of stochastic gradient updates. Each stochastic gradient update uses one of the observations (A_t, b_t, M_t), $t = 1, 2, \ldots, T$. Recall that the observations form an i.i.d. sequence satisfying Assumption 2.2. We order these observations according to the epoch / time-step in which they are accessed: we denote $(A_{s,t}, b_{s,t}, M_{s,t})$ to indicate the observation received in the t-th time step, $t \in [T_s]$, of the s-th epoch. The update directions are then obtained, as remarked after Lemma 3.1, using the estimated gradients of $L(\theta, y)$ at the current observation $(A_{s,t}, b_{s,t}, M_{s,t})$, given by:

$$g_{s,t}^y = \nabla_y L_{s,t}(\theta_{s,t}, y_{s,t}) = b_{s,t} - A_{s,t}\theta_{s,t} - M_{s,t}y_{s,t},$$

and

$$g_{s,t}^\theta = \nabla_\theta L_{s,t}(\theta_{s,t}, y_{s,t}) = -{A_{s,t}}^\top y_{s,t} .$$

Algorithm 1 Unconstrained algorithm with restarting

Input: arbitrary initial points $\theta_{1,1}$, $y_{1,1}$; initial step sizes η_1^θ, η_1^y; initial epoch length T_1.
for $s = 1$ to S **do**
  for $t = 1$ to T_s **do**
    $\theta_{s,t+1} \leftarrow \theta_{s,1} + \frac{1}{\eta_s^\theta} g_{s,1:t}^\theta$
    $y_{s,t+1} \leftarrow y_{s,1} + \frac{1}{\eta_s^y} g_{s,1:t}^y$
  end for
  $\theta_{(s+1),1} \leftarrow \bar{\theta}_s := \frac{1}{T_s}\sum_{t=1}^{T_s} \theta_{s,t}$
  $y_{(s+1),1} \leftarrow 0$
  $\eta_{s+1}^\theta \leftarrow 2\eta_s^\theta$, $\eta_{s+1}^y \leftarrow 2\eta_s^y$, and $T_{s+1} \leftarrow 2T_s$
end for
return $\bar{\theta}_S = \frac{1}{T_S}\sum_{t=1}^{T_S} \theta_{S,t}$

In each epoch (Line 4 and Line 5), we perform standard primal-dual updates via the FTRL algorithm. At the end of the epoch, we obtain the averaged primal variable $\bar{\theta}_s$ (Line 7) and set the dual variable $\bar{y}_s$ to zero (Line 8), which act as initial iterates for the next epoch.[8] Finally, in Line 9, the epoch length is doubled and the learning rates are halved for the next epoch.

Theorem 1. Suppose that the iterates θ_t and y_t, $t \in [T]$, are given by Algorithm 1. Then, for large enough $T_1 \approx \Theta(\frac{1}{\mu^2\beta^2})$, for step sizes $\eta^\theta_s = \frac{\mu T_s}{16}$ and $\eta^y_s = \frac{\beta T_s}{4}$, under Assumptions 2.1 and 2.2, after $S > 0$ epochs, we have $\ell(\bar{\theta}_S) - \ell(\theta^*) = \Theta(\frac{1}{\beta T_{1:S}})$.

Proof Sketch. The detailed proof is given in the extended version of the paper (Raj et al., 2022). The result directly comes from combining the bounds on the norm terms appearing in the regret bound of the primal, dual and noise variables, which have been provided in Lemmas 3.2, 3.3, 3.4 and 3.5. The negative Bregman divergences coming from the regret decomposition are used with the appropriate choice of the step size to cancel the positive norm terms arising from the regret bounds. Defining $\Delta\ell_s = \ell(\bar{\theta}_s) - \ell(\theta^*)$, we use strong convexity of $\ell(\theta)$ to get the recursion $\mathbb{E}[\Delta\ell_s] \le \frac{1}{4}\mathbb{E}[\Delta\ell_{s-1}] + \frac{C}{T_{1:s}}$ for some constant C which depends on fixed quantities $\theta^\star, y^\star, \beta, \mu, \lambda_A$ and λ_M. Since $T_s = 2T_{s-1}$, we get $\Delta\ell_s = \mathcal{O}(\frac{1}{T})$ where T is total number of iterations. □

5 IMPROVED RESULTS WITH BOUNDED Θ IN THE I.I.D. SETTING

In this section we make the extra assumption that we are given a set Θ with bounded diameter $\sup_{\theta_1,\theta_2\in\Theta} \|\theta_1 - \theta_2\|^2 \le r_\theta^2$, such that the projection $\mathcal{P}_\Theta(\theta)$ can be computed efficiently for any $\theta \in \mathbb{R}^d$. We design algorithms that keep their iterates inside Θ.

Besides the case when a constraint set Θ is given as part of the problem, such projected-update algorithms can also be used as a proxy to find the *unconstrained* solution $\theta^* \in \mathbb{R}^d$ of problem (1). To that end, we would need to create a projection-friendly set Θ that contains the unconstrained solution θ^*. Assuming A and M are invertible, such a constraint set can be built using extra prior knowledge. In particular, since $A\theta^* = b$, we have $\theta^* = A^{-1}b = (A^\top M^{-1}A)^{-1}(A^\top M^{-1}b)$. Hence, θ^* will be inside the ℓ_2-ball of radius r_θ as long as $r_\theta > \frac{\lambda_A}{\beta\mu}\|b\|$ or $r_\theta > \|b\|/\sigma_{\min}(A)$.

In previous work, the assumption of a bounded Θ has been commonly accompanied by the assumption that the dual iterate y_t is also projected to a convex set $\mathcal{Y}$. The results in this section show that this is not necessary, but can be beneficial. Specifically, when λ_A/β is known, projecting the dual variable to a $\mathcal{Y}$ containing the ℓ_2-ball of radius $\frac{\lambda_A}{\beta}r_\theta$ ensures that $\bar{y}^* \in \mathcal{Y}$ as required by Lemma 3.1; we can thus use a projected adaptive y_t update to relax the need to know λ_M.

Robust bounds. We start with robust bounds that do not require the knowledge of β. A proof is given in the extended version of the paper (Raj et al., 2022).

Theorem 2. Suppose θ_t and y_t are given by any of the three update pairs in Table 2. Then, under Assumptions 2.1 and 2.2, for all $T \ge 1$, the error of each pair of updates is upper-bounded as presented in Table 2. In all cases, $\ell(\bar{\theta}) - \ell(\theta^*) = \mathcal{O}(1/\sqrt{T})$.

Remark. Notably, the pair of updates in the second and third rows of Table 2 converge for *any* positive values of η^θ and η^y. These two parameters can still be further tuned using the standard hyper-parameter tuning methods and / or further prior knowledge, resulting in better constants in the convergence rate. This should be contrasted with the restricted range of step-sizes under which prior work typically establish convergence.

Adaptive $\mathcal{O}(1/T)$ bounds with knowledge of β. Finally, we show that given knowledge of β, one can make the same prior-knowledge trade-offs as the robust case, while still obtaining an $\mathcal{O}(1/T)$ rate. The next theorem formalizes this idea. A proof is provided in the extended version of the paper (Raj et al., 2022).

Theorem 3. Suppose θ_t and y_t are given by any of the three update pairs in Table 3. Then, under Assumptions 2.1 and 2.2, for all $T \ge 1$, $\ell(\bar{\theta}) - \ell(\theta^*) = \mathcal{O}(1/T)$.

6 MARKOV NOISE SETTING

In this section, we provide convergence guarantees for problem (1) in the more realistic Markov noise setting.

First, in Theorem 4, we consider the simpler case when both the primal variable θ and the dual variable y are projected onto the compact sets Θ and $\mathcal{Y}$, respectively.

Theorem 4. Suppose Assumptions 2.1 and 2.3 hold. Assume that $\sup_{\theta\in\Theta}\|\theta\|^2 \le r_\theta^2$ and $\sup_{y\in\mathcal{Y}}\|y\|^2 \le r_y^2$. Let $y_1 = y^* = 0$ and θ_1 be arbitrary, and for all $t \in \mathbb{N}$,

$$\theta_{t+1} = \mathcal{P}_\Theta\left[\theta_t - \frac{g^\theta_t}{\eta^\theta}\right], \text{ and } y_{t+1} = \mathcal{P}_\mathcal{Y}\left[y_t + \frac{g^y_t}{\eta^y_t}\right],$$

where $\eta^\theta = \frac{2\tau(\tau+1)\lambda_A^2}{\beta}$ and $\eta^y_t = \frac{4}{\beta t}$. Then, under Assumptions 2.1 and 2.3, for all $t \in [T]$ and $\Delta > 0$,

$$\ell(\bar{\theta}_T) - \ell(\theta^*) \le \frac{C_1\tau(\tau+1)}{\beta T} + \frac{10L_2^2 + C_2\tau}{\beta T}\log(1+T) + 2\Delta(r_\theta L_1 + r_y L_2) + \frac{6\tau r_y(\lambda_A r_\theta + L_2)}{T},$$

[8] For the generalized problem where A is not invertible, we can start epoch $s+1$ at $y_{s+1,1} = \bar{y}_s = \frac{1}{T_s}y_{s,1:T_s}$.

Update rule	Step-size	Bound on the Unnormalized Error: $T \cdot \mathbb{E}\left[\ell(\bar{\theta}_T) - \ell(\theta^*)\right]$
$\theta_{t+1} = \mathcal{P}_\Theta\left(\theta_t - g_t^\theta/\eta^\theta\right)$ $y_{t+1} = y_1 + g_{1:t}^y/\eta_t^y$	$\eta^\theta = 2\lambda_A^2/\beta$ $\eta_t^y = 12\lambda_M + \sqrt{t+1}$	$\frac{\lambda_A^2}{\beta} r_\theta^2 + (12\lambda_M + \sqrt{T})\frac{\lambda_A^2}{\beta^2} r_\theta^2 + 6\left(\sigma_*^2 + \lambda_A^2 r_\theta^2\right)\sqrt{T}$
$\theta_{t+1} = \mathcal{P}_\Theta\left(\theta_t - g_t^\theta/\eta_t^\theta\right)$ $y_{t+1} = y_1 + \frac{g_{1:t}^y}{12\lambda_M + \eta_t^y}$	$\eta_t^\theta = \eta^\theta\sqrt{\sum_{s=1}^{t-1}\|g_s^\theta\|^2}$ $\eta_t^y = \eta^y\sqrt{\sum_{s=1}^{t-1}\|g_s^y\|^2}$	$2\left(\frac{\eta^y\lambda_A^2}{2\beta^2} r_\theta^2 + \frac{1}{\eta^y}\right)\sqrt{3\sigma_*^2 T + 3\lambda_A^2 r_\theta^2 T} + \frac{2\lambda_A^2}{\beta}\left(\frac{\eta^\theta}{2} r_\theta^2 + \frac{1}{\eta^\theta}\right)^2$ $+24\lambda_M\left(\frac{\eta^y\lambda_A^2}{2\beta^2} r_\theta^2 + \frac{1}{\eta^y}\right)^2 + 12\lambda_M\frac{\lambda_A^2}{\beta^2} r_\theta^2 + \frac{\left(B^2 + \lambda_A^2 r_\theta^2\right)}{4\lambda_M}$
$\theta_{t+1} = \mathcal{P}_\Theta\left(\theta_t - g_t^\theta/\eta_t^\theta\right)$ $y_{t+1} = \mathcal{P}_\mathcal{Y}\left(y_t + g_t^y/\eta_t^y\right)$	$\eta_t^\theta = \eta^\theta\sqrt{\sum_{s=1}^{t-1}\|g_s^\theta\|^2}$ $\eta_t^y = \eta^y\sqrt{\sum_{s=1}^{t-1}\|g_s^y\|^2}$	$2\left(\frac{\eta^y c\lambda_A^2}{2\beta^2} r_\theta^2 + \frac{1}{\eta^y}\right)\sqrt{3\sigma_*^2 T + 3\lambda_A^2 r_\theta^2 T} + \frac{2\lambda_A^2}{\beta}\left(\frac{\eta^\theta}{2} r_\theta^2 + \frac{1}{\eta^\theta}\right)^2$ $+24\lambda_M\left(\frac{\eta^y c\lambda_A^2}{2\beta^2} r_\theta^2 + \frac{1}{\eta^y}\right)^2$

Table 2: Algorithm configurations that result in a robust $\mathcal{O}(1/\sqrt{T})$ convergence guarantee for bounded Θ. All update rules apply for $t \in [T]$. In all cases, $\theta_1 \in \Theta$ is arbitrary, and $y_1 = y^* = 0$. Note that the adaptive step sizes may be zero before the first non-zero gradient estimate is observed. In that case, the possible 0/0 in the update rule is evaluated to 0 by convention: the algorithm keeps predicting $\theta_{t+1} = \theta_1$ until it receives a non-zero update direction. In the last update, $\mathcal{Y}$ is any closed convex set containing the ℓ_2-ball of radius $\lambda_A^2 r_\theta^2/\beta^2$ satisfying $\sup_{y\in\mathcal{Y}}\|y\|^2 \le c\lambda_A^2 r_\theta^2/\beta^2$ for some $c \ge 1$.

Update rule	Step-size
$\theta_{t+1} = \mathcal{P}_\Theta\left(\theta_t - g_t^\theta/\eta^\theta\right)$ $y_{t+1} = y_t + g_t^y/\eta_t^y$	$\eta^\theta = 2\lambda_A^2/\beta$ $\eta_t^y = 16\lambda_M + \frac{\beta}{2}t$
$\theta_{t+1} = \mathcal{P}_\Theta\left(\theta_t - g_t^\theta/\eta_t^\theta\right)$ $y_{t+1} = y_t + g_t^y/\eta_t^y$	$\eta_t^\theta = \eta^\theta\sqrt{\sum_{s=1}^{t-1}\|g_s^\theta\|^2}$ $\eta_t^y = 16\lambda_M + \frac{\beta}{2}t$
$\theta_{t+1} = \mathcal{P}_\Theta\left(\theta_t - g_t^\theta/\eta_t^\theta\right)$ $y_{t+1} = \mathcal{P}_\mathcal{Y}\left(y_t + g_t^y/\eta_t^y\right)$	$\eta_t^\theta = \eta^\theta\sqrt{\sum_{s=1}^{t-1}\|g_s^\theta\|^2}$ $\eta_t^y = \frac{\beta}{2}t$

Table 3: Algorithm configurations that result in a fast $\mathcal{O}(1/T)$ convergence guarantee for bounded Θ. All update rules apply for $t \in [T]$. In all cases, $\theta_1 \in \Theta$ is arbitrary, and $y_1 = y^* = 0$. Similar comments as in Table 2 apply to adaptive steps and $\mathcal{Y}$.

where $L_1 = \lambda_A r_y$, $L_2 = B + \lambda_A r_\theta + \lambda_m r_y$, $C_1 = 2\lambda_A^2 r_\theta^2$, $C_2 = 4\left[2r_y\lambda_M L_2 + 2r_\theta\lambda_A L_2 + 3L_2^2\right]$, and $\tau = \tau_0(\Delta)$.

The result obtained above illuminates the effect of Markov noise on the convergence guarantee, and exposes the problem parameters appearing in the rate. Next, in Theorem 5, we consider the unbounded setting $\Theta = \mathcal{Y} = \mathbb{R}^d$ and analyze Algorithm 1 under Markov noise, proving a counterpart of Theorem 1.

Theorem 5. Suppose Assumptions 2.1 and 2.3 hold. Assume that the iterates θ_t and y_t are given by Algorithm 1 with step sizes $\eta_\theta^{(s)} = \Theta\left(\max\left\{\mu T_{1:s}, \frac{\beta T_{1:s}}{\lambda_A^2}\right\}\right)$ and $\eta_y^{(s)} = \Theta\left(\frac{\beta T_{1:s}}{\lambda_A^2}\right)$. Then, after $S > 0$ epochs, for small enough $\Delta = \mathcal{O}(\beta^2\mu^2)$, and for large enough $T_1 = \Omega\left(\frac{\tau}{\beta^2\mu^2}\right)$, we have

$$\ell(\bar{\theta}_S) - \ell(\theta^*) = \Theta\left(\frac{\tau}{\beta T_{1:S}} + \Delta\right).$$

7 IMPLICATIONS FOR GRADIENT TD LEARNING

In this section, we recall the formal reduction of the off-policy policy evaluation problem to problem (1), and discuss the implications of our results for this setting.

Following the notations in Liu et al. (2015), consider a Markov Decision Process (MDP), which is a tuple $\left(\mathcal{S}, \mathcal{A}, (\mathcal{P}_{ss'}^a)_{s,s'\in\mathcal{S},a\in\mathcal{A}}, R, \gamma\right)$, where $\mathcal{S}$ and $\mathcal{A}$, respectively, are the finite sets of states and actions, $\mathcal{P}_{ss'}^a$ is the probability of transition from state $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$ when performing action $a \in \mathcal{A}$, $R : S \times A \to [0, R_{max}]$ is the reward function with $R(s,a)$ denoting the reward received at state s if the agent performs action a, and $0 \le \gamma < 1$ is a discount factor. A stationary policy $\pi : S \times A \to [0,1]$, where $\sum_{a\in\mathcal{A}}\pi(s,a) = 1$ for every $s \in \mathcal{S}$, indicates the probability of performing a particular action at a particular state. The value function of a given policy π is denoted by $V^\pi : S \to R$ which happens to be the unique fixed-point of the Bellman operator T^π, that is,

$$V^\pi = T^\pi V^\pi = R^\pi + \gamma P^\pi V^\pi.$$

We consider the off-policy evaluation problem, where the goal is to estimate the value of a policy π, while the MDP can only be observed through interactions via another policy π_b, called the behavior policy. For a state-action pair (s,a), such that $\pi_b(a|s) > 0$, the importance weighting factor is defined as $\rho(s,a) = \frac{\pi(a|s)}{\pi_b(a|s)}$ with $\rho_{\max} \ge \sup_{s\in\mathcal{S},a\in\mathcal{A}}\rho(s,a)$ being the maximum. For large state spaces, we often approximate the value function V^π with linear functions as $\hat{v} = \Phi\theta$ where $\Phi \in \mathbb{R}^{|S|\times d}$ and $\theta \in \mathbb{R}^d$. We assume to have access to a stream of training samples which we denote as $\{(s_t, a_t, r_t = r(s_t,a_t), s_t', \rho_t)\}_{t=1}^\infty$, where r_t denotes

the reward after choosing action a_t at the state s_t, $s_t \sim P_t$ where P_t is the distribution of s_t at step t, $a_t \sim \pi_b(\cdot|s_t)$, $s'_t \sim P(\cdot|s_t, a_t)$ and $\rho_t = \rho(s_t, a_t)$. In the online (Markov noise) setting, we have $s'_t = s_{t+1}$. Alternatively, given a finite batch of n transitions $\{(s_i, a_i, r_i = r(s_i, a_i), s'_i, \rho(s_i, a_i))\}_{i=1}^n$, we can create the stream above under the i.i.d. noise setting by sampling from the batch in an i.i.d. fashion.

We use $\phi_t = \phi(s_t)$ to denote the feature vector corresponding to state s_t, and denote $\phi(s'_t)$ with ϕ'_t. Further, denote the TD error by $\delta_t(\theta) = r_t + \gamma {\phi_t'}^\top \theta - \phi_t^\top \theta$, and define $\Delta\phi_t = \phi_t - \gamma\phi'_t$. Then, the goal of GTD methods (Maei, 2011) is to solve problem (1) with

$$A = \lim_{t\to\infty} \mathbb{E}[A_t]\,,$$
$$b = \lim_{t\to\infty} \mathbb{E}[b_t]\,,$$
$$C = \lim_{t\to\infty} \mathbb{E}[C_t]\,,$$

where $A_t = \rho_t\phi_t\Delta\phi_t^\top$, $b_t = \rho_t\phi_t r_t$, and $C_t = \phi_i\phi_i^\top$. In this case, $\ell(\theta)$ of problem (1) is called the Mean Squared Projected Bellman Error (MSPBE) objective.

Interpretation of results for the MSPBE objective. Given the definitions above, it is straightforward to see that an upper-bound λ_M on $\|M_t\|$ can be immediately obtained as $\lambda_M = \sup_t \|\phi_t\|$. Thus, Theorem 2 shows that we can adaptively tune the GTD2 algorithm, using only knowledge of the feature norms, and obtain a convergence guarantee of $\mathcal{O}(1/\sqrt{T})$. Similarly, note that the minimum eigenvalue of C, lower-bounded by β, is independent of the tagert policy π. Hence, Theorem 3 implies a $\mathcal{O}(\frac{1}{\beta T})$ bound with proper adaptive tuning of GTD2. In both cases, the step sizes are independent of the potentially large factor $\rho_{\max}$; this would not be the case if we had to tune the step size based, e.g., on upper-bounds of the gradient norms obtained from λ_A or μ.

8 CONCLUSIONS

In this paper, we studied the problem of linear composition optimization, a special case of which arises in policy evaluation for reinforcement learning using gradient temporal difference learning algorithms. Applying a simple alternative to the saddle-point formulation of Liu et al. (2018), we exploited the structure (specifically, the curvature) of this optimization problem to achieve convergence rates for the objective that: a) remove or relax unnecessary assumptions about availability of prior knowledge, b) apply to adaptive, simple-to-tune algorithms, and c) enjoy better dependence on the problem parameters compared to previous work. In particular, we analyzed an unconstrained restarting scheme that, under both i.i.d. and Markov noise, achieves a $\mathcal{O}(1/(\beta T))$ rate, which to our knowledge has not been available in previous work. Finally, we discussed the implications of these results applied to the adaptive tuning of GTD algorithm for policy evaluation. In particular, we noted that the new step-size schedules do not depend on the largest importance sampling ratio $\rho_{\max}$, preventing the aggressive scale-down of the step-size that would potentially occur when the algorithm is tuned based on other problem parameters.

The next immediate research problem is to extend the analysis to non-linear gradient temporal difference learning, to value iteration, and to universal adaptive step-size algorithms that do not require *any* prior knowledge to converge in the unconstrained setting.

Acknowledgements

During part of this work, A. Raj was a research intern at DeepMind.

Bibliography

Francis Bach and Eric Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate o (1/n). In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 1*, pages 773–781, 2013.

Gal Dalal, Balázs Szörényi, Gugan Thoppe, and Shie Mannor. Finite sample analyses for td (0) with function approximation. *arXiv preprint arXiv:1704.01161*, 2017.

Gal Dalal, Gugan Thoppe, Balázs Szörényi, and Shie Mannor. Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning. In *Conference On Learning Theory*, pages 1199–1233. PMLR, 2018.

Gal Dalal, Balazs Szorenyi, and Gugan Thoppe. A tale of two-timescale reinforcement learning with the tightest finite-time bound. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3701–3708, 2020.

Thinh T Doan. Finite-time analysis and restarting scheme for linear two-time-scale stochastic approximation. *SIAM Journal on Control and Optimization*, 59(4):2798–2819, 2021.

Simon S Du, Jianshu Chen, Lihong Li, Lin Xiao, and Dengyong Zhou. Stochastic variance reduction methods for policy evaluation. In *International Conference on Machine Learning*, pages 1049–1058. PMLR, 2017.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(Jul):2121–2159, 2011.

Harsh Gupta, R Srikant, and Lei Ying. Finite-time performance bounds and adaptive learning rate selection for two time-scale reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Elad Hazan. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

Bin Hu and Usman Ahmed Syed. Characterizing the exact behaviors of temporal difference learning algorithms using markov jump linear system theory. *arXiv preprint arXiv:1906.06781*, 2019.

Kevin Huang and Shuzhong Zhang. New first-order algorithms for stochastic variational inequalities. *arXiv preprint arXiv:2107.08341*, 2021.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

Pooria Joulani, András György, and Csaba Szepesvári. A modular analysis of adaptive (non-) convex optimization: Optimism, composite objectives, variance reduction, and variational bounds. *Theoretical Computer Science*, 808:108–138, 2020.

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

Maxim Kaledin, Eric Moulines, Alexey Naumov, Vladislav Tadic, and Hoi-To Wai. Finite time analysis of linear two-timescale stochastic approximation with markovian noise. *arXiv preprint arXiv:2002.01268*, 2020.

Georgios Kotsalis, Guanghui Lan, and Tianjiao Li. Simple and optimal methods for stochastic variational inequalities, i: operator extrapolation. *arXiv preprint arXiv:2011.02987*, 2020a.

Georgios Kotsalis, Guanghui Lan, and Tianjiao Li. Simple and optimal methods for stochastic variational inequalities, ii: Markovian noise and policy evaluation in reinforcement learning. *arXiv preprint arXiv:2011.08434*, 2020b.

Bo Liu, Ji Liu, Mohammad Ghavamzadeh, Sridhar Mahadevan, and Marek Petrik. Finite-sample analysis of proximal gradient td algorithms. In *UAI*, pages 504–513. Citeseer, 2015.

Bo Liu, Ian Gemp, Mohammad Ghavamzadeh, Ji Liu, Sridhar Mahadevan, and Marek Petrik. Proximal gradient temporal difference learning: Stable reinforcement learning with polynomial sample complexity. *Journal of Artificial Intelligence Research*, 63: 461–494, 2018.

Hamid Reza Maei. *Gradient temporal-difference learning algorithms*. PhD thesis, University of Alberta, 2011.

H Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization. *arXiv preprint arXiv:1002.4908*, 2010.

Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

Zilun Peng, Ahmed Touati, Pascal Vincent, and Doina Precup. Svrg for policy evaluation with fewer gradient evaluations. *arXiv preprint arXiv:1906.03704*, 2019.

Anant Raj, Pooria Joulani, András György, and Csaba Szepesvári. Faster rates, adaptive algorithms, and finite-time bounds for linear composition optimization and gradient td learning. *arXiv preprint*, 2022.

Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International conference on machine learning*, pages 71–79. PMLR, 2013.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

Richard S Sutton, Csaba Szepesvári, and Hamid Reza Maei. A convergent o (n) algorithm for off-policy temporal-difference learning with linear function approximation. *Advances in neural information processing systems*, 21(21):1609–1616, 2008.

Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 993–1000, 2009.

Mengdi Wang, Ethan X Fang, and Han Liu. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *Mathematical Programming*, 161(1-2):419–449, 2017a.

Mengdi Wang, Ji Liu, and Ethan X. Fang. Accelerating stochastic composition optimization. *Journal of Machine Learning Research*, 18(105):1–23, 2017b.

Yue Wang, Wei Chen, Yuting Liu, Zhi-Ming Ma, and Tie-Yan Liu. Finite sample analysis of the gtd policy evaluation algorithms in markov setting. *arXiv preprint arXiv:1809.08926*, 2018.

Tengyu Xu, Shaofeng Zou, and Yingbin Liang. Two time-scale off-policy td learning: Non-asymptotic analysis over markovian samples. In *Advances in*

Neural Information Processing Systems, pages 10634–10644, 2019.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.