Convergence Rates of Accelerated Markov Gradient Descent with Applications in Reinforcement Learning

Thinh T. Doan Lam M. Nguyen Nhan H. Pham Justin Romberg

June 8, 2020

Abstract

Motivated by broad applications in machine learning, we study the popular accelerated stochastic gradient descent (ASGD) algorithm for solving (possibly nonconvex) optimization problems. We characterize the finite-time performance of this method when the gradients are sampled from Markov processes, and hence biased and dependent from time step to time step; in contrast, the analysis in existing work relies heavily on the stochastic gradients being independent and sometimes unbiased. Our main contributions show that under certain (standard) assumptions on the underlying Markov chain generating the gradients, ASGD converges at the nearly the same rate with Markovian gradient samples as with independent gradient samples. The only difference is a logarithmic factor that accounts for the mixing time of the Markov chain.

One of the key motivations for this study are complicated control problems that can be modeled by a Markov decision process and solved using reinforcement learning. We apply the accelerated method to several challenging problems in the OpenAI Gym and Mujoco, and show that acceleration can significantly improve the performance of the classic REINFORCE algorithm.

1 Introduction

Stochastic gradient descent (SGD) and its variants, originally introduced in [29] under the name of stochastic approximation (SA), is the most efficient and widely used method for solving optimization problems in machine learning (RL) [3, 16, 6, 25] and reinforcement learning [30, 31]. It can substantially reduce the cost of computing a step direction in supervised learning, and offers a framework for systematically handling uncertainty in reinforcement learning. In this context, we want to optimize an (unknown) objective function f when queries for the gradient are noisy. At a point x, we observe a random vector $G(x, \xi)$ whose mean is

Thinh T. Doan, School of Electrical and Computer Engineering, Georgia Institute of Technology, GA, USA. Email: thinhdoan@gatech.edu

Lam M. Nguyen, IBM Research, Thomas J. Watson Research Center, Yorktown Heights, NY, USA. Email: LamNguyen.MLTD@ibm.com

Nhan H. Pham, Department of Statistics and Operations Research, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA. Email: nhanph@live.unc.edu

Justin Romberg, School of Electrical and Computer Engineering, Georgia Institute of Technology, GA, USA. Email: jrom@ece.gatech.edu

the (sub)gradient of f at x. Through judicious choice of step sizes, the "noise" induced by this randomness can be averaged out across iterations, and the algorithm converges to the stationary point of f [12, 3, 26].

To further improve the performance of SGD, stochastic versions of Nesterovs acceleration scheme [24] have been studied in different settings [14, 41, 23]. In many of these cases, it has been observed that acceleration improves the performance of SGD both in theory [12, 35, 7] and in practice [17], with a notable application in neural networks [1]. This benefit of accelerated SGD (ASGD) has been studied under the i.i.d noise settings. Almost nothing is known when the noise is Markovian, which is often considered in the context of RL problems modeled by Markov decision processes [37].

In this paper, we show that a particular version of ASGD is still ergodic when the gradients of the objective are sampled from Markov process, and hence are biased and not independent across iterations. This model for the gradients has been considered previously in [11, 34, 15, 28], where different variants of SGD are considered. Moreover, it has also been observed that the SGD performs better when the gradients are sampled from Markov process as compared to i.i.d samples in both convex and nonconvex problems [34]. This paper shows that the benefits of acceleration extend to the Markovian setting in theory and in practice; we provide theoretical convergence rates that nearly match those in the i.i.d. setting, and show empirically that the algorithm is able to learn from significantly fewer samples on benchmark reinforcement learning problems.

Main contributions. We study accelerated stochastic gradient descent where the gradients are sampled from a Markov process. We show that, despite the gradients being biased and dependent across iterations, the convergence rate across many different types of objective functions (convex and smooth, strongly convex, nonconvex and smooth) is within a logarithmic factor of the comparable bounds for independent gradients. This logarithmic factor is naturally related to the mixing time of the underlying Markov process generating the stochastic gradients. To our knowledge, these are the first such bounds for accelerated stochastic gradient descent with Markovian sampling.

We also show that acceleration is extremely effective in practice by applying it to multiple problems in reinforcement learning. Compared with the popular temporal difference learning and Monte-Carlo policy gradient REINFORCE algorithms, the accelerated variants require significantly fewer samples to learn a policy with comparable rewards, which aligns with our theoretical results.

2 Accelerated Markov gradient descent

We consider the (possibly nonconvex) optimization problem

$$\underset{x\in\mathcal{X}}{\text{minimize}}\ f(x), \tag{1}$$

where $\mathcal{X}\subset\mathbb{R}^d$ is a closed convex set and $f:\mathcal{X}\to\mathbb{R}$ is given as

$$f(x)\triangleq\mathbb{E}_\pi[F(x;\xi)]=\int_\Xi F(x;\xi)d\pi(\xi). \tag{2}$$

Here Ξ is a statistical sample space with probability distribution π and $F(\cdot;\xi):\mathcal{X}\to\mathbb{R}$ is a bounded below (possibly nonconvex) function associated with $\xi\in\Xi$. We are interested in the first-order stochastic optimization methods for solving problem (1). Most of existing algorithms, such as SGD, require a sequence

of $\{\xi_k\}$ sampled i.i.d from the distribution π. Our focus is to consider the case where $\{\xi_k\}$ are generated from an ergodic Markov process, whose stationary distribution is π.

We focus on studying accelerated gradient methods for solving problem (1), originally proposed by Nesterov [24] and studied later in different variants; see for example [20, 12, 14, 41] and the reference therein. In particular, we study an ergodic version of ASGD studied in [20, 21, 12], where the gradients are sampled from a Markov process. We name this algorithm as accelerated Markov gradient descent formally stated in Algorithms 1 and 2 for nonconvex and convex problems, respectively. Our goal is to derive the rates of this method, which is unknown in the literature.

In our algorithms, $G(x;\xi) \in \partial F(x;\xi)$ is the subgradient of $F(\cdot;\xi)$ evaluated at x. As mentioned we consider the case where $\{\xi_k\}$ is drawn from a Markov ergodic stochastic process. We denote by $\tau(\gamma)$ the mixing time of the Markov chain $\{\xi_k\}$ given a positive constant γ, which basically tells us how long the Markov chain gets close to the stationary distribution [22]. To provide a finite-time analysis of this algorithm, we consider the following fairly standard assumption about the Markov process.

Assumption 1. *The Markov chain $\{\xi_k\}$ with finite state Ξ is ergodic, i.e., irreducible and aperiodic.*

Assumption 1 implies that $\{\xi_k\}$ has geometric mixing time[1] , i.e., given $\gamma > 0$ there exists $C > 0$ s.t.

$$\tau(\gamma) = C\log(1/\gamma) \quad \text{and} \quad \|\mathbb{P}^k(\xi,\cdot) - \pi\|_{TV} \leq \gamma, \quad \forall k \geq \tau(\gamma),\ \forall \xi \in \Xi, \tag{3}$$

where $\|\cdot\|_{TV}$ is the total variance distance [22]. This assumption holds in various applications, e.g, in incremental optimization [28], where the iterates are updated based on a finite Markov chain. Similar observation holds in reinforcement learning problems that have a finite number of states and actions, for example in AlphaGo [32]. Assumption 1 is used in the existing literature to study the finite-time performance of SA under Markov randomness; see [34, 33, 5, 8, 43, 9] and the references therein.

Before proceeding to the finite-time analysis of the accelerated Markov gradient descent, we present the motivation behind our approach and theoretical results given later. To study the asymptotic convergence of SGD under Markovian noise, one may use the popular ordinary differential equation (ODE) approach in stochastic approximation literature, where the geometric mixing time is unnecessary, see for example [2, 19]. However, we note that our focus is to study the finite-time performance of ASGD. The existing techniques in studying ASGD rely on the main assumptions that the gradients are sampled i.i.d from the (unknown) stationary distribution π and unbiased. In our setting, since the gradients are sampled from a Markov process, they are dependent and biased (nonstationary). Even if we can sample from π ($\tau = 0$ and the gradient samples are unbiased), they are still dependent. Thus, it is not trivial to handle the bias and dependence simultaneously using the existing techniques. We, therefore, utilize the geometric mixing time to eliminate this issue in our analysis. Indeed, under Assumption 1, we show that the convergence rates of the accelerated Markov gradient descent are the same with the ones in ASGD under i.i.d. samples for solving both convex and nonconvex problems, except for a $\log(k)$ factor which captures the mixing time of the underlying Markov chain.

3 Convergence analysis: Nonconvex case

We study Algorithm 1 for solving problem (1) when $\mathcal{X} = \mathbb{R}^d$, and f is nonconvex satisfying the assumptions below.

[1] τ depends on the second largest eigenvalue of the transition probability matrix of the Markov chain.

Algorithm 1 Accelerated Markov Gradient Descent

Initialize: Set arbitrarily $x_0, \bar{x}_0 \in \mathcal{X}$, step sizes $\{\alpha_k, \beta_k, \gamma_k\}$, and an integer $K \geq 1$

Iterations: For $k = 1, \ldots, K$ do

$$y_k = (1 - \alpha_k)\bar{x}_{k-1} + \alpha_k x_{k-1} \tag{4}$$
$$x_k = x_{k-1} - \gamma_k G(y_k; \xi_k) \tag{5}$$
$$\bar{x}_k = y_k - \beta_k G(y_k; \xi_k) \tag{6}$$

Output: y_R randomly selected from the sequence $\{y_k\}_{k=1}^K$ with probability p_k defined as

$$p_k = \frac{\gamma_k(1 - L\gamma_k)}{\sum_{k=1}^K \gamma_k(1 - L\gamma_k)}. \tag{7}$$

Assumption 2. $f^* = \inf_{x \in \mathbb{R}^d} f(x) > -\infty$.

In addition, we assume that ∇f and its samples are Lipschitz continuous and bounded as in [5, 34].

Assumption 3. *There exists a constant $L > 0$ such that $\forall x, y$ and $\forall \xi \in \Xi$*

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \qquad \text{and} \qquad \|G(x; \xi) - G(y; \xi)\| \leq L\|x - y\|. \tag{8}$$

Assumption 4. *There exists a constant $M > 0$ such that $\forall x$ and $\forall \xi \in \Xi$*

$$\max\{\|\nabla f(x)\|, \|G(x; \xi)\|\} \leq M. \tag{9}$$

In this section, we assume that Assumptions 1–4 always hold. Given α_k, let Γ_k be defined as

$$\Gamma_k = \begin{cases} 1, & k \leq 1 \\ (1 - \alpha_k)\Gamma_{k-1} & k \geq 2. \end{cases} \tag{10}$$

We first consider the following key lemma, which is essential in the analysis of Theorem 1 below.

Lemma 1. *Let $\{\gamma_k, \beta_k\}$ be nonnegative and nonincreasing and $\beta_k \leq \gamma_k$. Then*

$$\begin{aligned} \mathbb{E}[f(x_k)] \leq \mathbb{E}[f(x_{k-1})] - \gamma_k (1 - L\gamma_k) \mathbb{E}\left[\|\nabla f(y_k)\|^2\right] + 8LM^2\tau(\gamma_k)\gamma_{k-\tau(\gamma_k)}\gamma_k \\ + M^2 L \left(2M\gamma_k \sum_{t=k-\tau(\gamma_k)}^{k} \alpha_t\Gamma_t + \frac{\Gamma_k}{2}\right) \sum_{t=1}^{k} \frac{(\gamma_t - \beta_t)^2}{\Gamma_t \alpha_t} + (4M^2L + M)\gamma_k^2. \end{aligned} \tag{11}$$

Sketch of proof. A complete analysis of this lemma is presented in the supplementary material. Here, we briefly discuss the main technical challenge in our analysis due to Markov samples, that is, the gradient samples are biased and dependent. First, using Assumption 3 with some manipulation gives

$$\begin{aligned} f(x_k) \leq f(x_{k-1}) - \gamma_k (1 - L\gamma_k) \|\nabla f(y_k)\|^2 + \frac{M^2 L\Gamma_k}{2} \sum_{t=1}^{k} \frac{(\gamma_t - \beta_t)^2}{\Gamma_t \alpha_t} + 4M^2 L\gamma_k^2 \\ - \gamma_k \langle \nabla f(x_{k-1}),\, G(y_k; \xi_k) - \nabla f(y_k) \rangle. \end{aligned}$$

In the i.i.d settings, since the gradient samples are unbiased and independent, the last term on the right-hand side has a zero expectation. However, in our setting this expectation is different to zero and the samples are dependent. We, therefore, cannot apply the existing techniques to show (11). Our key technique to address this challenge is to utilize the geometric mixing time τ defined in (3). In particular, although the noise in our algorithm is Markovian, its dependence is very weak at samples spaced out at every τ step. We, therefore, carefully characterize the progress of the algorithm in every τ step, resulting to the sum of over τ steps on the right-hand side of (11). □

To show our result for smooth nonconvex problems, we adopt the randomized stopping rule in [12], which is common used in nonconvex optimization. In particular, given a sequence $\{y_k\}$ generated by Algorithm 1 we study the convergence on y_R, a point randomly selected from this sequence (a.k.a (13)). The convergence rate of Algorithm 1 in solving problem (1) is stated as follows.

Theorem 1. *Let $K > 0$ be an integer such that*

$$\alpha_k = \frac{2}{k+1}, \quad \gamma_k \in [\beta_k\,,\,(1+\alpha_k)\beta_k]\,, \quad \beta_k = \beta = \frac{1}{\sqrt{K}} \leq \frac{1}{4L}, \quad \forall k \geq 1. \tag{12}$$

In addition, let y_R be randomly selected from the sequence $\{y_k\}_{k=1}^K$ with probability p_k defined as

$$p_k = \frac{\gamma_k(1-L\gamma_k)}{\sum_{k=1}^K \gamma_k(1-L\gamma_k)}. \tag{13}$$

Then y_R returned by Algorithm 1 satisfies[2]

$$\mathbb{E}\left[\|\nabla f(y_R)\|^2\right] \leq \frac{2(\mathbb{E}[f(x_0)]-f^*)\left(4L+\sqrt{K}\right)}{K} + \frac{2M(LM^2(9+16C\log(K))+1+M^2)}{\sqrt{K}}. \tag{14}$$

Proof. Using (10) and (12) yields $\Gamma_k = 2/k(k+1)$. Thus, using the integral test we have

$$\gamma_k \sum_{t=k-\tau(\gamma_k)}^{k} \alpha_t\Gamma_t = \gamma_k \sum_{t=k-\tau(\gamma_k)}^{k} \frac{4}{t(t+1)^2} \leq \frac{2\gamma_k}{(k-\tau(\gamma_k))^2}. \tag{15}$$

Next, using (12) and $\Gamma_k = 2/k(k+1)$ we consider

$$\begin{aligned}
&\sum_{k=1}^K \Gamma_k \sum_{t=1}^k \frac{(\gamma_t-\beta_t)^2}{\Gamma_t\alpha_t} = \sum_{t=1}^K \frac{(\gamma_t-\beta_t)^2}{\Gamma_t\alpha_t}\sum_{k=t}^K \Gamma_k = \sum_{t=1}^K \frac{(\gamma_t-\beta_t)^2}{\Gamma_t\alpha_t}\sum_{k=t}^K \frac{2}{k(k+1)} \\
&= \sum_{t=1}^K \frac{2(\gamma_t-\beta_t)^2}{\Gamma_t\alpha_t}\sum_{k=t}^K \left(\frac{1}{k}-\frac{1}{k+1}\right) \leq \sum_{t=1}^K \frac{2(\gamma_t-\beta_t)^2}{\Gamma_t\alpha_t}\frac{1}{t} \leq 2\sum_{t=1}^K \frac{\beta_t^2\alpha_t^2}{t\Gamma_t\alpha_t} = 2\beta^2 K.
\end{aligned} \tag{16}$$

Similarly, using (15), $\alpha_k \leq 1$, $\gamma_k \leq 2\beta_k = 2\beta$ we have

$$\begin{aligned}
&\sum_{k=1}^K \gamma_k \sum_{t=k-\tau(\gamma_k)}^{k} \alpha_t\Gamma_t \sum_{t=1}^k \frac{(\gamma_t-\beta_t)^2}{\Gamma_t\alpha_t} \leq \sum_{k=1}^K \frac{2\gamma_k}{(k-\tau(\gamma_k))^2}\sum_{t=1}^k \frac{(\gamma_t-\beta_t)^2}{\Gamma_t\alpha_t} \\
&\leq \sum_{t=1}^K \frac{2\beta^3\alpha_t^2}{\Gamma_t\alpha_t}\sum_{k=t}^K \frac{1}{(k-\tau(\gamma_k))^2} \leq 2\beta^3\sum_{t=1}^K \frac{t}{t-\tau(\gamma_t)} \leq 4\beta^3 K.
\end{aligned} \tag{17}$$

[2]Note that the same rate can be achieved for the quantity $\min_k \mathbb{E}\left[\|\nabla f(y_k)\|^2\right]$.

Moreover, using (3) and $\gamma_k \geq \beta_k = \beta$ we have $\tau(\gamma_k) = C \log\left(\frac{1}{\gamma_k}\right) \leq C \log(1/\beta)$, which gives

$$\sum_{k=1}^{K} \tau(\gamma_k)\gamma_{k-\tau(\gamma_k)}\gamma_k \leq 4C \sum_{k=1}^{K} \beta_k \beta_{k-\tau(\gamma_k)} \log(k) = 2C\beta^2 K \log(K). \tag{18}$$

Since $\alpha_k \leq 1$ for $k \geq 1$ we have $\sum_{k=1}^{K} \gamma_k^2 \leq 2\beta^2 K$. We now use the relations (16)–(18) to derive (14). Indeed, summing up both sides of (11) over k from 1 to N and reorganizing yield

$$\begin{aligned}
&\sum_{k=1}^{K} \gamma_k \left(1 - L\gamma_k\right) \mathbb{E}\left[\|\nabla f(y_k)\|^2\right] \\
&\leq \mathbb{E}[f(x_0)] - \mathbb{E}[f(x_K)] + (4LM^2 + M) \sum_{k=1}^{K} \gamma_k^2 + 8LM^2 \sum_{k=1}^{K} \tau(\gamma_k)\gamma_{k-\tau(\gamma_k)}\gamma_k \\
&\quad + \sum_{k=1}^{K} \frac{M^2 L \Gamma_k}{2} \sum_{t=1}^{k} \frac{(\gamma_t - \beta_t)^2}{\Gamma_t \alpha_t} + 2M^3 L \sum_{k=1}^{K} \gamma_k \sum_{t=k-\tau(\gamma_k)}^{k} \alpha_t \Gamma_t \sum_{t=1}^{k} \frac{(\gamma_t - \beta_t)^2}{\Gamma_t \alpha_t} \\
&\leq (\mathbb{E}[f(x_0)] - f^*) + (9LM^2 + 2M + M^3)\beta^2 K + 16CLM^2\beta^2 K \log(K),
\end{aligned} \tag{19}$$

where we use $\mathbb{E}[f(x_K)] \geq f^*$ and $\beta \leq 1/4L$. Dividing both sides by $\sum_{k=1}^{K} \gamma_k(1 - L\gamma_k)$ gives

$$\begin{aligned}
\frac{\sum_{k=1}^{K} \gamma_k \left(1 - L\gamma_k\right) \mathbb{E}\left[\|\nabla f(y_k)\|^2\right]}{\sum_{k=1}^{K} \gamma_k(1 - L\gamma_k)} &\leq \frac{(\mathbb{E}[f(x_0)] - f^*)}{\sum_{k=1}^{K} \gamma_k(1 - L\gamma_k)} + \frac{(9LM^2 + 2M + M^3)\beta^2 K}{\sum_{k=1}^{K} \gamma_k(1 - L\gamma_k)} \\
&\quad + \frac{16CLM^2\beta^2 K \log(K)}{\sum_{k=1}^{K} \gamma_k(1 - L\gamma_k)}.
\end{aligned}$$

Using (12) yields $1 - L\gamma_k \geq 1/2$ and $\sum_{k=1}^{K} \gamma_k(1 - L\gamma_k) \geq \sum_{k=1}^{K} \beta_k/2 = K\beta/2$. Thus, we obtain

$$\begin{aligned}
&\frac{\sum_{k=1}^{K} \gamma_k \left(1 - L\gamma_k\right) \mathbb{E}\left[\|\nabla f(y_k)\|^2\right]}{\sum_{k=1}^{K} \gamma_k(1 - L\gamma_k)} \\
&\leq \frac{2(\mathbb{E}[f(x_0)] - f^*)}{K\beta} + \frac{2(9LM^2 + 2M + M^3)\beta^2 K}{K\beta} + \frac{32CLM^2\beta^2 K \log(K)}{K\beta} \\
&\leq \frac{2(\mathbb{E}[f(x_0)] - f^*)\left(4L + \sqrt{K}\right)}{K} + \frac{2(9LM^2 + 2M + M^3)}{\sqrt{K}} + \frac{32CLM^2 \log(K)}{\sqrt{K}},
\end{aligned}$$

which by using (13) gives (14). □

4 Convergence analysis: Convex case

In this section, we study Algorithm 2 for solving (1) when f is convex and $\mathcal{X}$ is compact.

For simplicity we consider V in Algorithm 2 is the Euclidean distance, i.e., $\psi(x) = \frac{1}{2}\|x\|^2$ and $V(y, x) = \frac{1}{2}\|y - x\|^2$. Since $\mathcal{X}$ is compact, there exist $D, M > 0$ s.t.

$$D = \max_{x \in \mathcal{X}} \|x\|, \quad \|G(x; \xi)\| \leq M, \quad \forall \xi \in \Xi, \ \forall x \in \mathcal{X}. \tag{23}$$

Algorithm 2 Accelerated Markov Gradient Descent

Initialize: Set arbitrarily $x_0, \bar{x}_0 \in \mathcal{X}$, step sizes $\{\alpha_k, \beta_k, \gamma_k\}$, and an integer $K \geq 1$

Iterations: For $k = 1, \ldots, K$ do

$$y_k = (1 - \beta_k)\bar{x}_{k-1} + \beta_k x_{k-1} \tag{20}$$

$$x_k = \arg\min_{x \in \mathcal{X}} \left\{ \gamma_k \left[\langle G(y_k; \xi_k), x - y_k \rangle + \mu V(y_k, x) \right] + V(x_{k-1}, x) \right\} \tag{21}$$

$$\bar{x}_k = (1 - \alpha_k)\bar{x}_{k-1} + \alpha_k x_k \tag{22}$$

Output: $\bar{x}_k$

In addition, let $x^* = \arg\min_{x \in \mathcal{X}} f(x)$. We assume that $\{\alpha_k, \beta_k\}$ are chosen such that $\alpha_1 = 1$ and

$$\frac{\beta_k(1 - \alpha_k)}{\alpha_k(1 - \beta_k)} = \frac{1}{1 + \mu\gamma_k}, \qquad 1 + \mu\gamma_k > L\alpha_k\gamma_k, \tag{24}$$

where $\mu \geq 0$ and L is given in (8). The key idea to derive the results in this section is to utilize Assumption 1 to the handle the Markovian "noise", similar to the one in Section 3. For an ease of exposition, we present the analysis of the results in this section to the supplementary material.

4.1 Smooth convex functions

We now study the rates of Algorithm 2 when the function f is only convex and Assumption 3 holds. In this case $\partial f(\cdot) = \nabla f(\cdot)$ and $\mu = 0$. Since $\mu = 0$, Eq. (24) gives $\beta_k = \alpha_k$ and $y_k = \bar{x}_k$. The convergence rate of Algorithm 2 in this case is given below.

Theorem 2. *Let Assumptions 1–3 hold. Suppose that the step sizes are chosen as*

$$\alpha_k = \frac{2}{k+1}, \quad \gamma_k = \frac{1}{2L\sqrt{k+1}}. \tag{25}$$

Then we have for all $k \geq 1$

$$f(\bar{x}_k) - f(x^*) \leq \frac{f(\bar{x}_0) + 4LD}{2k(k+1)} + \frac{2(D + 2M^2)}{3L\sqrt{k}} + \frac{2(4D^2L + M^2)(L+1)\log(2L\sqrt{k})}{\sqrt{k}}. \tag{26}$$

4.2 Strongly convex functions

We now provide the rates of Algorithm 2 when f is strongly convex.

Assumption 5. *There exists a constant $\mu > 0$ s.t. $\forall x, y$ and $g(x) \in \partial f(x)$ we have*

$$\frac{\mu}{2}\|y - x\|^2 \leq f(y) - f(x) - \langle g(x), y - x \rangle. \tag{27}$$

Theorem 3. *Suppose that Assumptions 1, 3, and 5 hold. Consider the step sizes chosen as*

$$\alpha_k = \frac{2}{k+1}, \quad \gamma_k = \frac{2}{\mu(k+1)}, \quad \beta_k = \frac{\alpha_k}{\alpha_k + (1 - \alpha_k)(1 + \mu\gamma_k)}. \tag{28}$$

Then we have for all $k \geq 1$

$$f(\bar{x}_k) - f(x^*) \leq \frac{2f(\bar{x}_0) + 6\mu D}{k(k+1)} + \frac{2D + 10M^2 + 8\mu MD}{\mu(k+1)} \tag{29}$$

$$+ \frac{4(M^2 + 2\mu MD + 12\mu LD^2)(2+\mu)\log(\frac{\mu(k+1)}{2})}{\mu k}. \tag{30}$$

Remark 1. *We note that in Theorem 2,* ASGD *has the same worst case convergence rate as compared to* SGD*, i.e.,* $\mathcal{O}\, 1/\sqrt{k}$*. However,* ASGD *has much better rate on the initial condition than* SGD*, i.e.,* $\mathcal{O}(1/k^2)$ *versus* $\mathcal{O}(1/k)$*. Similar observation holds for Theorem 3. This gain is very important, for example, in improving the data efficiency of RL algorithms as illustrated in Section 5.1.*

5 Numerical experiments

In this section, we apply the proposed accelerated Markov gradient methods for solving a number of problems in reinforcement learning, where the samples are taken from Markov processes. In particular, we consider the usual setup of reinforcement learning where the environment is modeled by a Markov decision process (MDP) [37]. Let $\mathcal{S}$ and $\mathcal{A}$ be the (finite) set of states and action. We denote by $\pi_\theta(s,a) = Pr(a_k = a | s_k = s, \theta)$ the randomized policy parameterized by θ, where $s \in \mathcal{S}$ and $a \in \mathcal{A}$. The goal is to find θ to maximize the cumulative reward

$$f(\pi_\theta) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_k \,|\, s_0, \pi_\theta\right],$$

where γ is the discounted factor and r_k is the reward returned by the environment at time k. We study the accelerated variants of temporal difference learning and Monte-Carlo policy gradient (REINFORCE) methods, and compare their performance with the classic (non-accelerated) counterparts. In all the experiments we consider below, the proposed accelerated variants of RL algorithms outperform the classic ones, which agrees with our theoretical results.

General setup: For each simulation, we run the algorithm 10 times with the same initial policy and record the performance measures. The performance of each algorithm is specified by averaging the metric over the number of episodes. Here, an episode is defined as the set of state-action pairs collected from beginning until the terminal state or a specified episode is reached. The plots consist of the mean with 90% confidence interval (shaded area) of the performance metric. For REINFORCE-based methods, we randomly generate an initial policy represented by a neural network.

5.1 Accelerated temporal difference learning

One of the central problems in RL is the so-called policy evaluation problem, that is, we want to estimate $f(\pi_\theta)$ for a fixed policy π_θ. Temporal difference learning TD(λ), originally proposed by Sutton [36], is one of the most efficient and practical methods for policy evaluation. It is shown in [27] that if the underlying Markov process is reversible, TD learning is a gradient descent method. In addition, under linear function approximation the TD method can be viewed as gradient descent for solving a strongly convex quadratic problem [40]. In this problem, the data tuple $\{s_k, a_k, s_{k+1}\}$ generated by the MDP is ξ_k in our model; see [40] for more details.

For our simulation, we consider the policy evaluation problem over the GridWorld environment [37, Example 4.1], where the agent is placed in a grid and wants to reach a goal from an initial position. The starting and goal positions are fixed at the top-left and bottom-right corners, respectively. We implement the one-step TD (or TD(0)), and apply our framework to obtain its accelerated variant, denoted as TD(0)-Acc. The value function is approximated by using linear function approximation, i.e., $f_\theta(s) = \langle \theta, \Phi(s) \rangle$ where $\Phi(s)$ is the feature at $s \in \mathcal{S}$ using $\mathcal{O}(3)$ order Fourier basis [18]. We consider a randomized policy choosing action uniformly over the set $\{up, down, left, right\}$. In this case, the transition matrix is doubly stochastic, therefore, reversible with uniform distribution.

Since the optimal solution is unknown, we use the norm of expected TD update (NEU) as in [38] to compare the performance of TD(0) and TD(0)-Acc. In each run, after every 10 episodes, the NEU is computed by averaging over 10 test episodes. The performance of both TD(0) variants the gridworld environment with size 10×10 and 50×50 are presented in Figure 1, which shows that the proposed method, TD(0)-Acc, outperforms the classic TD(0). The detail of parameter selection of this simulation is presented in the supplementary material.

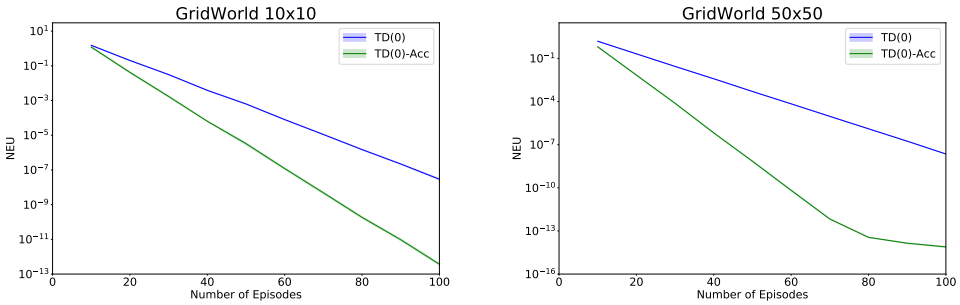

Figure 1: The performance of two TD(0) variants on GridWorld with sizes 10×10 (left) and 50×50 (right).

5.2 Accelerated REINFORCE methods

The REINFORCE method can be viewed as SGD in reinforcement learning [42]. To evaluate the two variants of REINFORCE, we consider five different control problems, namely, Acrobot, CartPole, Ant, Swimmer, and HalfCheetah, using the simulated environments from OpenAI Gym and Mujoco [4, 39] . We utilize the implementation of REINFORCE from `rllab` library [10]. More details of these environments and simulations in this section are given in the supplementary material.

Brief summary: At every iteration, we collect a batch of episodes with different length depending on the environment. We then update the policy parameters and record the performance measure by collecting 50 episodes using the updated policy and average the total rewards for all episodes.

We first compare the algorithms using discrete control tasks: `Acrobot-v1` and `CartPole-v0` environments. For these discrete tasks, we use a soft-max policy π_θ with parameter θ defined as

$$\pi_\theta(a|s) = \frac{e^{\phi_(s,a,\theta)}}{\sum_{k=1}^{|\mathcal{A}|} e^{\phi(s,a_k,\theta)}}, \tag{31}$$

where $\phi(s, a, \theta)$ is represented by a neural network and $|\mathcal{A}|$ is the total number of actions. Figure 2 presents

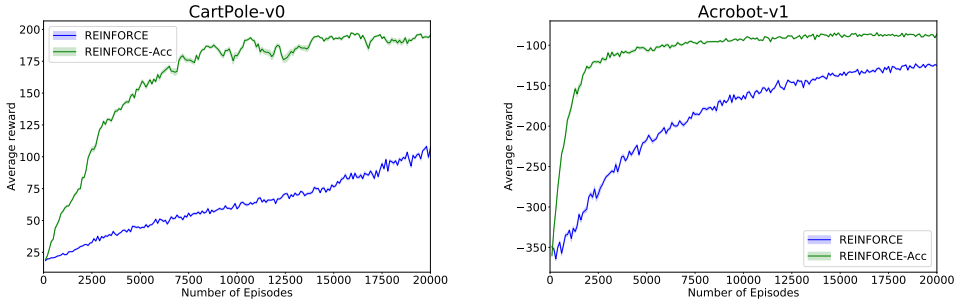

Figure 2: The performance of two algorithms on the `CartPole-v0` and `Acrobot-v1` environment.

the performance of two algorithms on these environments. In both environments, the accelerated REINFORCE significantly outperforms its non-accelerated variant.

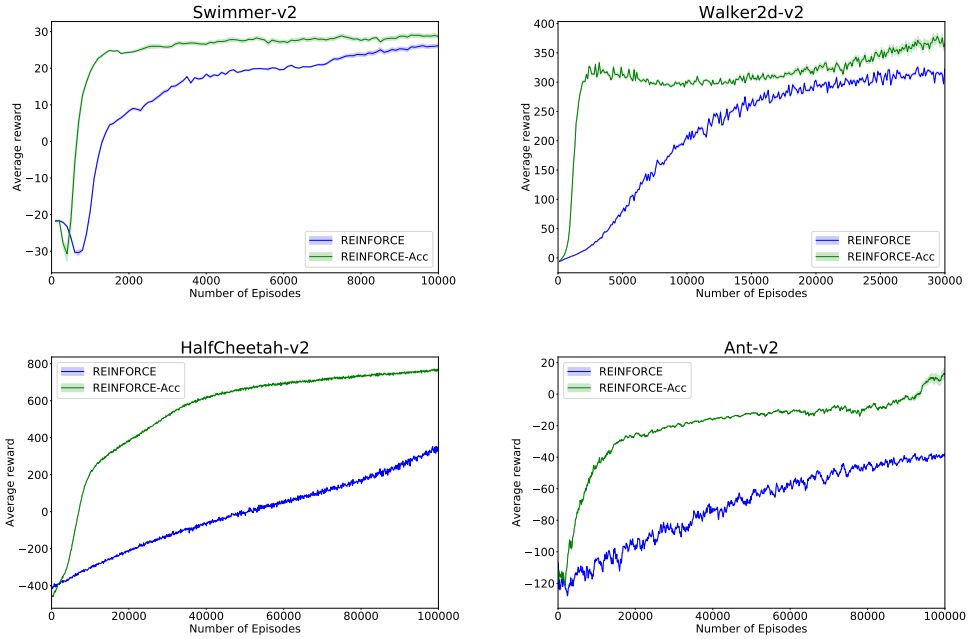

Figure 3: The performance of two algorithms on 4 Mujoco environments.

Next, we evaluate the performance of these algorithms on continuous control tasks in Mujoco. In these environments, we also incorporate a linear baseline to reduce the variance of the policy gradient estimator, see [13]. The actions are sampled from a deep Gaussian policy which can be written as $\pi_\theta(a|s) = \mathcal{N}(\phi(s,a,\theta_\mu); \phi(s,a,\theta_\sigma))$, where $\phi(.)$ is a neural network. The mean and variance of the Gaussian distribution is learned in this experiment.

We evaluate these algorithms on four environments with increasing difficulty: `Swimmer`, `Walker2d`, `HalfCheetah`, and `Ant`. Figure 3 illustrates the results in those environments, respectively. In all figures, REINFORCE-Acc indeed shows its advantage over REINFORCE.

6 Conclusion

In this paper, we study a variant of ASGD for solving (possibly nonconvex) optimization problems, when the gradients are sampled from Makrov process. We characterize the finite-time performance of this method when the gradients are sampled from Markov processes, which shows that ASGD converges at the nearly the same rate with Markovian gradient samples as with independent gradient samples. The only difference is a logarithmic factor that accounts for the mixing time of the Markov chain. We apply the accelerated methods to policy evaluation problems in GridWorld environment and to several challenging problems in the OpenAI Gym and Mujoco. Our simulations show that acceleration can significantly improve the performance of the classic RL algorithms.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] V.S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.

[3] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

[4] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym, 2016.

[5] C. Zaiwei Chen, S. Zhang, T. T. Doan, S. T. Maguluri, and J-P. Clarke. Performance of Q-learning with Linear Function Approximation: Stability and Finite-Time Analysis. available at: `https://arxiv.org/abs/1905.11425`, 2019.

[6] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*, pages 1646–1654, 2014.

[7] Aymeric Dieuleveut, Nicolas Flammarion, and Francis Bach. Harder, better, faster, stronger convergence rates for least-squares regression. *The Journal of Machine Learning Research*, 18(1):3520–3570, 2017.

[8] T. T. Doan. Finite-time analysis and restarting scheme for linear two-time-scale stochastic approximation. available at: `https://arxiv.org/abs/1912.10583`, 2019.

[9] T. T. Doan, S. T. Maguluri, and J. Romberg. Finite-time performance of distributed temporal difference learning with linear function approximation. available at: `https://arxiv.org/abs/1907.12530`, 2019.

[10] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML16, page 13291338. JMLR.org, 2016.

[11] J. C. Duchi, A. Agarwal, M. Johansson, and M.I. Jordan. Ergodic mirror descent. *SIAM Journal on Optimization*, 22(4):1549–1578, 2012.

[12] S. Ghadimi and G. Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[13] Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.

[14] Prateek Jain, Sham M Kakade, Rahul Kidambi, Praneeth Netrapalli, and Aaron Sidford. Accelerating stochastic gradient descent for least squares regression. *arXiv preprint arXiv:1704.08227*, 2017.

[15] B. Johansson, M. Rabi, and M. Johansson. A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM Journal on Optimization*, 20(3):1157–1170, 2010.

[16] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, pages 315–323, 2013.

[17] Hiroyuki Kasai. Sgdlibrary: A matlab library for stochastic optimization algorithms. *The Journal of Machine Learning Research*, 18(1):7942–7946, 2017.

[18] G. Konidaris, S. Osentoski, and P. Thomas. Value function approximation in reinforcement learning using the fourier basis. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI11, page 380385. AAAI Press, 2011.

[19] H. J. Kushner and G. G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, New York, 2003.

[20] G. Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1):365–397, Jun 2012.

[21] G. Lan. Lectures on optimization methods for machine learning, 2019.

[22] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov chains and mixing times*. American Mathematical Society, 2006.

[23] Chaoyue Liu and Mikhail Belkin. Accelerating sgd with momentum for over-parameterized learning. *arXiv preprint arXiv:1810.13395*, 2018.

[24] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. 1983.

[25] Lam M. Nguyen, Jie Liu, Katya Scheinberg, and Martin Takac. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning*, pages 2613–2621, 2017.

[26] Lam M. Nguyen, Phuong Ha Nguyen, Peter Richtárik, Katya Scheinberg, Martin Takáč, and Marten van Dijk. New convergence aspects of stochastic gradient algorithms. *Journal of Machine Learning Research*, 20(176):1–49, 2019.

[27] Y. Ollivier. Approximate temporal difference learning is a gradient descent for reversible policies. available at: `https://arxiv.org/abs/1805.00869`, 2018.

[28] S.S. Ram, A. Nedi, and V. V. Veeravalli. Incremental stochastic subgradient algorithms for convex optimization. *SIAM Journal on Optimization*, 20(2):691–717, 2009.

[29] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[30] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.

[31] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[32] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[33] R. Srikant and L. Ying. Finite-time error bounds for linear stochastic approximation and TD learning. In *COLT*, 2019.

[34] T. Sun, Y. Sun, and W. Yin. On markov chain gradient descent. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS18, page 99189927, Red Hook, NY, USA, 2018. Curran Associates Inc.

[35] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[36] R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, Aug 1988.

[37] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning, 2nd Edition*. MIT Press, 2018.

[38] R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML 09, page 9931000, New York, NY, USA, 2009. Association for Computing Machinery.

[39] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[40] J. N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997.

[41] Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of sgd for over-parameterized models and an accelerated perceptron. *arXiv preprint arXiv:1810.07288*, 2018.

[42] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, May 1992.

[43] Yue Wu, Weitong Zhang, Pan Xu, and Quanquan Gu. A finite time analysis of two time-scale actor critic methods. *arXiv preprint arXiv:2005.01350*, 2020.

A Appendix

We first state the following result, as a consequence of the geometric mixing time in Assumptions 1 and Lipschitz continuity in 3 and 4. The proof of this result can be found in [5, Lemma 3.2], therefore, it is omitted here for brevity.

Corollary 1. *Suppose that Assumptions 1, 3, 4 hold. Let $g(x) \in \partial f(x)$ and $\tau(\gamma)$ defined in* (3)*. Then*

$$\|\mathbb{E}[G(x;\xi_k)] - g(x) \,|\, \xi_0 = \xi\| \leq \gamma, \quad \forall x,\ \forall k \geq \tau(\gamma). \tag{32}$$

A.1 Proof of Lemma 1

Proof. Since f satisfies Assumption 3, by (6) and (5) we have

$$\begin{aligned}
f(x_k) &\leq f(x_{k-1}) + \langle \nabla f(x_{k-1}),\, x_k - x_{k-1}\rangle + \frac{L}{2}\|x_k - x_{k-1}\|^2 \\
&= f(x_{k-1}) - \gamma_k \langle \nabla f(x_{k-1}),\, G(y_k;\xi_k)\rangle + \frac{L\gamma_k^2}{2}\|G(y_k;\xi_k)\|^2 \\
&= f(x_{k-1}) - \gamma_k \langle \nabla f(x_{k-1}),\, \nabla f(y_k)\rangle - \gamma_k \langle \nabla f(x_{k-1}),\, G(y_k;\xi_k) - \nabla f(y_k)\rangle \\
&\quad + \frac{L\gamma_k^2}{2}\|G(y_k;\xi_k)\|^2 \\
&= f(x_{k-1}) - \gamma_k \|\nabla f(y_k)\|^2 - \gamma_k \langle \nabla f(x_{k-1}) - \nabla f(y_k),\, \nabla f(y_k)\rangle \\
&\quad - \gamma_k \langle \nabla f(x_{k-1}),\, G(y_k;\xi_k) - \nabla f(y_k)\rangle + \frac{L\gamma_k^2}{2}\|G(y_k;\xi_k) - \nabla f(y_k) + \nabla f(y_k)\|^2 \\
&= f(x_{k-1}) - \gamma_k \left(1 - \frac{L\gamma_k}{2}\right)\|\nabla f(y_k)\|^2 - \gamma_k \langle \nabla f(x_{k-1}) - \nabla f(y_k),\, \nabla f(y_k)\rangle \\
&\quad - \gamma_k \langle \nabla f(x_{k-1}) - L\gamma_k \nabla f(y_k),\, G(y_k;\xi_k) - \nabla f(y_k)\rangle + \frac{L\gamma_k^2}{2}\|G(y_k;\xi_k) - \nabla f(y_k)\|^2 \\
&\leq f(x_{k-1}) - \gamma_k \left(1 - \frac{L\gamma_k}{2}\right)\|\nabla f(y_k)\|^2 + L\gamma_k \|x_{k-1} - y_k\|\|\nabla f(y_k)\| \\
&\quad - \gamma_k \langle \nabla f(x_{k-1}) - L\gamma_k \nabla f(y_k),\, G(y_k;\xi_k) - \nabla f(y_k)\rangle + 2M^2 L\gamma_k^2,
\end{aligned}$$

where the last inequality is due to 8 and (9). Using (6) we have from the preceding relation

$$\begin{aligned}
f(x_k) &\leq f(x_{k-1}) - \gamma_k \left(1 - \frac{L\gamma_k}{2}\right)\|\nabla f(y_k)\|^2 + L(1-\alpha_k)\gamma_k \|x_{k-1} - \bar{x}_{k-1}\|\|\nabla f(y_k)\| \\
&\quad - \gamma_k \langle \nabla f(x_{k-1}) - L\gamma_k \nabla f(y_k),\, G(y_k;\xi_k) - \nabla f(y_k)\rangle + 2M^2 L\gamma_k^2 \\
&\leq f(x_{k-1}) - \gamma_k \left(1 - L\gamma_k\right)\|\nabla f(y_k)\|^2 + \frac{L(1-\alpha_k)^2}{2}\|x_{k-1} - \bar{x}_{k-1}\|^2 \\
&\quad - \gamma_k \langle \nabla f(x_{k-1}) - L\gamma_k \nabla f(y_k),\, G(y_k;\xi_k) - \nabla f(y_k)\rangle + 2M^2 L\gamma_k^2,
\end{aligned} \tag{33}$$

where the last inequality we apply the relation $2ab \leq a^2 + b^2$ to the third term. Next, using (4)–(6) we have

$$\bar{x}_k - x_k = (1-\alpha_k)(\bar{x}_{k-1} - x_{k-1}) + (\gamma_k - \beta_k) G(y_k;\xi_k),$$

which dividing both sides by Γ_k, using (10) and $\alpha_1 = 1$, and summing up both sides yields

$$\bar{x}_k - x_k = \Gamma_k \sum_{t=1}^{k} \frac{\gamma_t - \beta_t}{\Gamma_t} G(y_t; \xi_t).$$

Thus, by using the Jensen's inequality for $\|\cdot\|^2$ and (48) we have from the preceding equation

$$\begin{aligned}\|\bar{x}_k - x_k\|^2 &= \left\|\Gamma_k \sum_{t=1}^{k} \frac{\gamma_t - \beta_t}{\Gamma_t} G(y_t; \xi_t)\right\|^2 = \left\|\Gamma_k \sum_{t=1}^{k} \frac{\alpha_t}{\Gamma_t} \frac{\gamma_t - \beta_t}{\alpha_t} G(y_t; \xi_t)\right\|^2 \\ &\leq \Gamma_k \sum_{t=1}^{k} \frac{\alpha_t}{\Gamma_t} \left\|\frac{\gamma_t - \beta_t}{\alpha_t} G(y_t; \xi_t)\right\|^2 \leq M^2 \Gamma_k \sum_{t=1}^{k} \frac{(\gamma_t - \beta_t)^2}{\Gamma_t \alpha_t}, \end{aligned} \tag{34}$$

where the last inequality is due to (9). Substituing the preceding relation into (33) and since $(1-\alpha_k)^2 \Gamma_{k-1} \leq \Gamma_k$ we have

$$\begin{aligned} f(x_k) &\leq f(x_{k-1}) - \gamma_k (1 - L\gamma_k) \|\nabla f(y_k)\|^2 + \frac{M^2 L \Gamma_k}{2} \sum_{t=1}^{k} \frac{(\gamma_t - \beta_t)^2}{\Gamma_t \alpha_t} \\ &\quad - \gamma_k \langle \nabla f(x_{k-1}) - L\gamma_k \nabla f(y_k),\, G(y_k; \xi_k) - \nabla f(y_k) \rangle + 2M^2 L \gamma_k^2 \\ &\leq f(x_{k-1}) - \gamma_k (1 - L\gamma_k) \|\nabla f(y_k)\|^2 + \frac{M^2 L \Gamma_k}{2} \sum_{t=1}^{k} \frac{(\gamma_t - \beta_t)^2}{\Gamma_t \alpha_t} \\ &\quad - \gamma_k \langle \nabla f(x_{k-1}),\, G(y_k; \xi_k) - \nabla f(y_k) \rangle + 4M^2 L \gamma_k^2, \end{aligned} \tag{35}$$

where the last inequality is due to (9). We next analyze the inner product on the right-hand side of (35). Indeed, by Assumptions 3 and 4 we have

$$\begin{aligned} &- \gamma_k \langle \nabla f(x_{k-1}),\, G(y_k; \xi_k) - \nabla f(y_k) \rangle \\ &= -\gamma_k \langle \nabla f(x_{k-\tau(\gamma_k)}),\, G(y_k; \xi_k) - \nabla f(y_k) \rangle \\ &\quad - \gamma_k \langle \nabla f(x_{k-1}) - \nabla f(x_{k-\tau(\gamma_k)}),\, G(y_k; \xi_k) - \nabla f(y_k) \rangle \\ &= -\gamma_k \langle \nabla f(x_{k-\tau(\gamma_k)}),\, G(y_{k-\tau(\gamma_k)}; \xi_k) - \nabla f(y_{k-\tau(\gamma_k)}) \rangle \\ &\quad - \gamma_k \langle \nabla f(x_{k-\tau(\gamma_k)}),\, G(y_k); \xi_k) - G(y_{k-\tau(\gamma_k)}; \xi_k) \rangle \\ &\quad - \gamma_k \langle \nabla f(x_{k-\tau(\gamma_k)}),\, \nabla f(y_{k-\tau(\gamma_k)}) - \nabla f(y_k) \rangle \\ &\quad - \gamma_k \langle \nabla f(x_{k-1}) - \nabla f(x_{k-\tau(\gamma_k)}),\, G(y_k; \xi_k) - \nabla f(y_k) \rangle \\ &\leq -\gamma_k \langle \nabla f(x_{k-\tau(\gamma_k)}),\, G(y_{k-\tau(\gamma_k)}; \xi_k) - \nabla f(y_{k-\tau(\gamma_k)}) \rangle + 2ML\gamma_k \|y_k - y_{k-\tau(\gamma_k)}\| \\ &\quad + 2LM\gamma_k \|x_{k-1} - x_{k-\tau(\gamma_k)}\|. \end{aligned} \tag{36}$$

First, we denote by $\mathcal{F}_k$ the filtration containing all the history generated by the algorithm up to time k. Using Eq. (32) we consider

$$\begin{aligned} &\mathbb{E}[-\gamma_k \langle \nabla f(x_{k-\tau(\gamma_k)}),\, G(y_{k-\tau(\gamma_k)}; \xi_k) - \nabla f(y_{k-\tau(\gamma_k)}) \rangle \,|\, \mathcal{F}_{k-\tau(\gamma_k)}] \\ &= -\gamma_k \langle \nabla f(x_{k-\tau(\gamma_k)}),\, \mathbb{E}[G(y_{k-\tau(\gamma_k)}; \xi_k) - \nabla f(y_{k-\tau(\gamma_k)}) \,|\, \mathcal{F}_{k-\tau(\gamma_k)}] \rangle \\ &\leq M\gamma_k \left| \mathbb{E}[G(y_{k-\tau(\gamma_k)}; \xi_k) - \nabla f(y_{k-\tau(\gamma_k)}) \,|\, \mathcal{F}_{k-\tau(\gamma_k)}] \right| \leq M\gamma_k^2. \end{aligned}$$

Second, by Eq. (6) we have

$$\begin{aligned} y_{k+1} - y_k &= y_{k+1} - \bar{x}_{k+1} + \bar{x}_{k+1} - \bar{x}_k + \bar{x}_k - y_k \\ &= \beta_{k+1} G(y_{k+1}; \xi_{k+1}) - \beta_k G(y_k; \xi_k) + \bar{x}_{k+1} - \bar{x}_k, \end{aligned}$$

which by Assumption 4 and since $\beta_{k+1} \leq \beta_k$ implies that

$$\|y_{k+1} - y_k\| \leq 2M\beta_k + \|\bar{x}_{k+1} - \bar{x}_k\|. \tag{37}$$

Using (4) and (6) we have

$$\bar{x}_{k+1} - \bar{x}_k = \alpha_{k+1}(x_k - \bar{x}_k) - \beta_{k+1}G(y_{k+1}; \xi_{k+1}),$$

which by using Eq. (34), $\alpha_{k+1} \leq \alpha_k$, and Assumption 4 implies that

$$\|\bar{x}_{k+1} - \bar{x}_k\| \leq M^2\alpha_k\Gamma_k \sum_{t=1}^{k} \frac{(\gamma_t - \beta_t)^2}{\Gamma_t\alpha_t} + M\beta_k.$$

Substituting the preceding relation into (37) yields

$$\|y_{k+1} - y_k\| \leq 3M\beta_k + M^2\alpha_k\Gamma_k \sum_{t=1}^{k} \frac{(\gamma_t - \beta_t)^2}{\Gamma_t\alpha_t},$$

which gives

$$\begin{aligned}
2LM\gamma_k\|y_k - y_{k-\tau(\gamma_k)}\| &\leq 2LM\gamma_k \sum_{t=k-\tau(\gamma_k)}^{k-1} \|y_{t+1} - y_t\| \\
&\leq 6LM^2\gamma_k \sum_{t=k-\tau(\gamma_k)}^{k-1} \beta_t + 2LM^3\gamma_k \sum_{t=k-\tau(\gamma_k)}^{k-1} \alpha_t\Gamma_t \sum_{u=1}^{t} \frac{(\gamma_u - \beta_u)^2}{\Gamma_u\alpha_u} \\
&\leq 6LM^2\tau(\gamma_k)\gamma_k\beta_{k-\tau(\gamma_k)} + 2LM^3\gamma_k \sum_{t=k-\tau(\gamma_k)}^{k-1} \alpha_t\Gamma_t \sum_{t=1}^{k} \frac{(\gamma_t - \beta_t)^2}{\Gamma_t\alpha_t} \\
&\leq 6LM^2\tau(\gamma_k)\gamma_{k-\tau(\gamma_k)}\gamma_k + 2LM^3\gamma_k \sum_{t=k-\tau(\gamma_k)}^{k-1} \alpha_t\Gamma_t \sum_{t=1}^{k} \frac{(\gamma_t - \beta_t)^2}{\Gamma_t\alpha_t}.
\end{aligned}$$

Third, we have

$$\begin{aligned}
2LM\gamma_k\|x_{k-1} - x_{k-\tau(\gamma_k)}\| &\leq 2LM\gamma_k \sum_{t=k-\tau(\gamma_k)+1}^{k-1} \|x_t - x_{t-1}\| \\
= 2LM\gamma_k \sum_{t=k-\tau(\gamma_k)+1}^{k-1} \|\gamma_t G(y_t; \xi_t)\| &\leq 2LM^2\tau(\gamma_k)\gamma_{k-\tau(\gamma_k)}\gamma_k.
\end{aligned}$$

Taking the expectation on both sides of (36) and using the preceding three relations we obtain

$$\begin{aligned}
&\mathbb{E}\left[-\gamma_k\langle\nabla f(x_{k-1}),\, G(y_k; \xi_k) - \nabla f(y_k)\rangle\right] \\
&\quad \leq M\gamma_k^2 + 8LM^2\tau(\gamma_k)\gamma_{k-\tau(\gamma_k)}\gamma_k + +2LM^3\gamma_k \sum_{t=k-\tau(\gamma_k)}^{k-1} \alpha_t\Gamma_t \sum_{t=1}^{k} \frac{(\gamma_t - \beta_t)^2}{\Gamma_t\alpha_t}.
\end{aligned}$$

Taking the expectation on both sides of (35) and using the equation above give (11). □

A.2 Proofs of Section 4

The analysis of Theorems 2 and 3 are established based on the following two key lemmas. The proof of the first lemma is adopted from the results studied in [20]. We restate here with some minor modification for the purpose of our analysis.

Lemma 2. *Let α_k and γ_k satisfy* (24) *and*

$$\frac{\alpha_k}{\gamma_k \Gamma_k} \leq \frac{\alpha_{k-1}(1+\mu\gamma_{k-1})}{\gamma_{k-1}\Gamma_{k-1}}, \tag{38}$$

where Γ_k is defined in (10)*. Then $\{\bar{x}_k\}$ generated by Algorithm 2 satisfies for all $k \geq 1$*

$$\begin{aligned} f(\bar{x}_k) - f(x^*) \leq \Gamma_k \frac{\gamma_0 f(\bar{x}_0) + \alpha_0(1+\mu\gamma_0)D}{\gamma_0 \Gamma_0} + \Gamma_k \sum_{t=1}^{k} \frac{4M^2 \gamma_t \alpha_t}{\Gamma_t(1+\mu\gamma_t - L\gamma_t\alpha_t)} \\ + \Gamma_k \sum_{t=1}^{k} \frac{\alpha_t}{\Gamma_t} \big\langle G(y_t;\xi_t) - \nabla f(y_t)\,,\, z - \tilde{x}_{t-1} \big\rangle, \end{aligned} \tag{39}$$

where $\tilde{x}_{k-1}$ is defined as

$$\tilde{x}_{k-1} = \frac{1}{1+\mu\gamma_k} x_{k-1} + \frac{\mu\gamma_k}{1+\mu\gamma_k} y_k. \tag{40}$$

Proof. Using the convexity of f, i.e.,

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle, \quad \forall x, y \in \mathbb{R}^d$$

and (22) we have for all $z \in \mathcal{X}$

$$\begin{aligned} f(z) + \langle \nabla f(z), \bar{x}_k - z \rangle &= f(z) + \langle \nabla f(z), \alpha_k x_k + (1-\alpha_k)\bar{x}_{k-1} - z \rangle \\ &= (1-\alpha_k)\big[f(z) + \langle \nabla f(z), \bar{x}_{k-1} - z \rangle\big] + \alpha_k \big[f(z) + \langle \nabla f(z), x_k - z \rangle\big] \\ &\leq (1-\alpha_k) f(\bar{x}_{k-1}) + \alpha_k \big[f(z) + \langle \nabla f(z), x_k - z \rangle\big]. \end{aligned}$$

By the preceding relation and (8) we have for all $z \in \mathcal{X}$

$$\begin{aligned} f(\bar{x}_k) &\leq f(z) + \langle \nabla f(z), \bar{x}_k - z \rangle + \frac{L}{2}\|\bar{x}_k - z\|^2 \\ &\leq (1-\alpha_k) f(\bar{x}_{k-1}) + \alpha_k \big[f(z) + \langle \nabla f(z), x_k - z \rangle\big] + \frac{L}{2}\|\bar{x}_k - z\|^2, \end{aligned}$$

which by letting $z = y_k$ we obtain

$$f(\bar{x}_k) \leq (1-\alpha_k) f(\bar{x}_{k-1}) + \alpha_k \big[f(y_k) + \langle \nabla f(y_k), x_k - y_k \rangle\big] + \frac{L}{2}\|\bar{x}_k - y_k\|^2. \tag{41}$$

By the update of x_k in (21) and Lemma 3.5 in [21] we have for all $z \in \mathcal{X}$

$$\begin{aligned} &\gamma_k \left[\big\langle G(y_k;\xi_k)\,,\, x_k - y_k \big\rangle + \mu V(y_k, x_k)\right] + V(x_{k-1}, x_k) \\ &\quad \leq \gamma_k \left[\big\langle G(y_k;\xi_k)\,,\, z - y_k \big\rangle + \mu V(y_k, z)\right] + V(x_{k-1}, z) - (1+\mu\gamma_k) V(x_k, z), \end{aligned}$$

which implies that

$$\begin{aligned}\left\langle \nabla f(y_k)\,,\, x_k - y_k \right\rangle &\leq \mu V(y_k, z) + \frac{1}{\gamma_k} V(x_{k-1}, z) - \frac{1+\mu\gamma_k}{\gamma_k} V(x_k, z) - \mu V(y_k, x_k) - \frac{1}{\gamma_k} V(x_{k-1}, x_k) \\ &\quad + \left\langle G(y_k;\xi_k)\,,\, z - y_k \right\rangle - \left\langle G(y_k;\xi_k) - \nabla f(y_k)\,,\, x_k - y_k \right\rangle \\ &= \mu V(y_k, z) + \frac{1}{\gamma_k} V(x_{k-1}, z) - \frac{1+\mu\gamma_k}{\gamma_k} V(x_k, z) - \mu V(y_k, x_k) - \frac{1}{\gamma_k} V(x_{k-1}, x_k) \\ &\quad + \left\langle \nabla f(y_k)\,,\, z - y_k \right\rangle + \left\langle G(y_k;\xi_k) - \nabla f(y_k)\,,\, z - x_k \right\rangle.\end{aligned}$$

Substituting the preceding equation into Eq. (41) yields

$$\begin{aligned}f(\bar{x}_k) &\leq (1-\alpha_k) f(\bar{x}_{k-1}) + \alpha_k \big[f(y_k) + \left\langle \nabla f(y_k)\,,\, z - y_k \right\rangle + \mu V(y_k, z) \big] + \frac{L}{2} \|\bar{x}_k - y_k\|^2 \\ &\quad + \frac{\alpha_k}{\gamma_k} \Big[V(x_{k-1}, z) - (1+\mu\gamma_k) V(x_k, z) - V(x_{k-1}, x_k) - \mu\gamma_k V(y_k, x_k) \Big] \\ &\quad + \alpha_k \left\langle G(y_k;\xi_k) - \nabla f(y_k)\,,\, z - x_k \right\rangle. \end{aligned} \tag{42}$$

We denote by $\tilde{x}_{k-1}$

$$\tilde{x}_{k-1} = \frac{1}{1+\mu\gamma_k} x_{k-1} + \frac{\mu\gamma_k}{1+\mu\gamma_k} y_k. \tag{43}$$

And note that

$$\frac{1}{1+\mu\gamma_k} = \frac{\beta_k(1-\alpha_k)}{\alpha_k(1-\beta_k)} \qquad \text{and} \qquad \frac{\mu\gamma_k}{1+\mu\gamma_k} = \frac{\alpha_k - \beta_k}{\alpha_k(1-\beta_k)}$$

Thus, using Eqs. (20) and (22), and the preceding relations we then have

$$\begin{aligned}\bar{x}_k - y_k &= \alpha_k x_k + \frac{1-\alpha_k}{1-\beta_k} (y_k - \beta_k x_{k-1}) - y_k = \alpha_k \left[x_k - \frac{\beta_k(1-\alpha_k)}{\alpha_k(1-\beta_k)} x_{k-1} - \frac{\alpha_k - \beta_k}{\alpha_k(1-\beta_k)} y_k \right] \\ &= \alpha_k (x_k - \tilde{x}_{k-1}). \end{aligned} \tag{44}$$

On the other hand, using the strong convexity of V we have

$$\begin{aligned}V(x_{k-1}, x_k) + \mu\gamma_k V(y_k, x_k) &\geq \frac{1}{2}\|x_{k-1} - x_k\|^2 + \frac{\mu\gamma_k}{2}\|x_k - y_k\|^2 \\ &\geq \frac{1+\mu\gamma_k}{2} \left\| \frac{1}{1+\mu\gamma_k}(x_k - x_{k-1}) + \frac{\mu\gamma_k}{1+\mu\gamma_k}(x_k - y_k) \right\|^2 \\ &= \frac{1+\mu\gamma_k}{2} \left\| x_k - \frac{1}{1+\mu\gamma_k} x_{k-1} - \frac{\mu\gamma_k}{1+\mu\gamma_k} y_k \right\|^2 \\ &= \frac{1+\mu\gamma_k}{2\alpha_k^2} \left\| \alpha_k (x_k - \tilde{x}_{k-1}) \right\|^2. \end{aligned} \tag{45}$$

We denote by $\Delta_k = G(y_k;\xi_k) - \nabla f(y_k)$. Then we consider

$$\begin{aligned}\left\langle G(y_k;\xi_k) - \nabla f(y_k)\,,\, z - x_k \right\rangle &= \left\langle G(y_k;\xi_k) - \nabla f(y_k)\,,\, \tilde{x}_{k-1} - x_k \right\rangle + \left\langle G(y_k;\xi_k) - \nabla f(y_k)\,,\, z - \tilde{x}_{k-1} \right\rangle \\ &\leq \|\Delta_k\| \|x_k - \tilde{x}_{k-1}\| + \left\langle \Delta_k\,,\, z - \tilde{x}_{k-1} \right\rangle. \end{aligned} \tag{46}$$

Substituting Eqs. (44)–(46) into Eq. (42) yields

$$\begin{aligned}
f(\bar{x}_k) &\le (1-\alpha_k) f(\bar{x}_{k-1}) + \alpha_k \big[f(y_k) + \big\langle \nabla f(y_k)\,,\, z - y_k \big\rangle + \mu V(y_k, z) \big] \\
&\quad + \frac{\alpha_k}{\gamma_k} \Big[V(x_{k-1}, z) - (1+\mu\gamma_k) V(x_k, z) \Big] + \alpha_k \big\langle G(y_k;\xi_k) - \nabla f(y_k)\,,\, z - \tilde{x}_{k-1} \big\rangle \\
&\quad - \left(\frac{1+\mu\gamma_k}{2\gamma_k\alpha_k} - \frac{L}{2} \right) \|\alpha_k (x_k - \tilde{x}_{k-1})\|^2 + \|\Delta_k\| \|\alpha_k (x_k - \tilde{x}_{k-1})\| \\
&\le (1-\alpha_k) f(\bar{x}_{k-1}) + \alpha_k \big[f(y_k) + \big\langle \nabla f(y_k)\,,\, z - y_k \big\rangle + \mu V(y_k, z) \big] \\
&\quad + \frac{\alpha_k}{\gamma_k} \Big[V(x_{k-1}, z) - (1+\mu\gamma_k) V(x_k, z) \Big] + \alpha_k \big\langle G(y_k;\xi_k) - \nabla f(y_k)\,,\, z - \tilde{x}_{k-1} \big\rangle \\
&\quad + \frac{\gamma_k \alpha_k \|\Delta_k\|^2}{1 + \mu\gamma_k - L\gamma_k\alpha_k}. \qquad (47)
\end{aligned}$$

Diving both sides of Eq. (47) by Γ_k and using (10) we have

$$\begin{aligned}
\frac{1}{\Gamma_k} f(\bar{x}_k) &\le \frac{1-\alpha_k}{\Gamma_k} f(\bar{x}_{k-1}) + \frac{\alpha_k}{\Gamma_k} \big[f(y_k) + \big\langle \nabla f(y_k)\,,\, z - y_k \big\rangle + \mu V(y_k, z) \big] + \frac{\gamma_k \alpha_k \|\Delta_k\|^2}{\Gamma_k (1 + \mu\gamma_k - L\gamma_k\alpha_k)} \\
&\quad + \frac{\alpha_k}{\Gamma_k \gamma_k} \Big[V(x_{k-1}, z) - (1+\mu\gamma_k) V(x_k, z) \Big] + \frac{\alpha_k}{\Gamma_k} \big\langle G(y_k;\xi_k) - \nabla f(y_k)\,,\, z - \tilde{x}_{k-1} \big\rangle \\
&\le \frac{1}{\Gamma_{k-1}} f(\bar{x}_{k-1}) + \frac{\alpha_k}{\Gamma_k} f(z) + \frac{\gamma_k \alpha_k \|\Delta_k\|^2}{\Gamma_k (1 + \mu\gamma_k - L\gamma_k\alpha_k)} \\
&\quad + \frac{\alpha_k}{\Gamma_k \gamma_k} \Big[V(x_{k-1}, z) - (1+\mu\gamma_k) V(x_k, z) \Big] + \frac{\alpha_k}{\Gamma_k} \big\langle G(y_k;\xi_k) - \nabla f(y_k)\,,\, z - \tilde{x}_{k-1} \big\rangle,
\end{aligned}$$

where the last inequality due to the convexity of f. Summing up both sides of the preceding relation over k from 1 to K yields

$$\begin{aligned}
f(\bar{x}_K) &\le \frac{\Gamma_K}{\Gamma_0} f(\bar{x}_0) + \Gamma_K \sum_{k=1}^{K} \frac{\alpha_k}{\Gamma_k} f(z) + \Gamma_K \sum_{k=1}^{K} \frac{\gamma_k \alpha_k \|\Delta_k\|^2}{\Gamma_k (1 + \mu\gamma_k - L\gamma_k\alpha_k)} \\
&\quad + \Gamma_K \sum_{k=1}^{K} \frac{\alpha_k}{\Gamma_k \gamma_k} \Big[V(x_{k-1}, z) - (1+\mu\gamma_k) V(x_k, z) \Big] \\
&\quad + \Gamma_K \sum_{k=1}^{K} \frac{\alpha_k}{\Gamma_k} \big\langle G(y_k;\xi_k) - \nabla f(y_k)\,,\, z - \tilde{x}_{k-1} \big\rangle \\
&\le \frac{\Gamma_K}{\Gamma_0} f(\bar{x}_0) + f(z) + \Gamma_K \sum_{k=1}^{K} \frac{\gamma_k \alpha_k \|\Delta_k\|^2}{\Gamma_k (1 + \mu\gamma_k - L\gamma_k\alpha_k)} \\
&\quad + \Gamma_K \left[\frac{\alpha_0 (1+\mu\gamma_0)}{\gamma_0 \Gamma_0} V(x_0, z) - \frac{\alpha_K (1+\mu\gamma_K)}{\Gamma_K \gamma_K} V(x_K, z) \right] \\
&\quad + \Gamma_K \sum_{k=1}^{K} \frac{\alpha_k}{\Gamma_k} \big\langle G(y_k;\xi_k) - \nabla f(y_k)\,,\, z - \tilde{x}_{k-1} \big\rangle,
\end{aligned}$$

where the second inequality is due to (38), $\alpha_1 = 1$, and the definition of Γ_k to have

$$\sum_{k=1}^{K} \frac{\alpha_k}{\Gamma_k} = \frac{\alpha_1}{\Gamma_1} + \sum_{k=2}^{K} \frac{1}{\Gamma_k} \left(1 - \frac{\Gamma_k}{\Gamma_{k-1}} \right) = \frac{1}{\Gamma_1} + \sum_{k=2}^{K} \left(\frac{1}{\Gamma_k} - \frac{1}{\Gamma_{k-1}} \right) = \frac{1}{\Gamma_K}. \qquad (48)$$

Thus, by letting $z = x^*$ in the preceding equation and since $\|\Delta_k\|^2 \le 4M^2$ we obtain (39). □

A.2.1 Proof of Theorem 2

To prove theorem 2, we first consider the following lemma, where handle the inner product on the right-hand side of (39) by using the geometric mixing time, similar to Lemma 1. Recall that since $\mu = 0$, we have $\beta_k = \alpha_k$ and $y_k = \bar{x}_k$. Thus, the updates in Algorithm 2 become

$$\bar{x}_k = (1-\alpha_k)\bar{x}_{k-1} + \alpha_k x_{k-1} \tag{49}$$

$$x_k = \arg\min_{x\in\mathcal{X}} \Big\{\gamma_k \langle G(\bar{x}_k;\xi_k)\,,\, x - \bar{x}_k\rangle + \frac{1}{2}\|x - x_{k-1}\|^2\Big\}. \tag{50}$$

Lemma 3. *Let the sequences $\{x_k, y_k\}$ be generated by Algorithm 2. Then*

$$\mathbb{E}[\langle G(\bar{x}_k;\xi_k) - \nabla f(\bar{x}_k), z - x_{k-1}\rangle] \le 2D\gamma_k + 2(4D^2L + M^2)\tau(\gamma_k)\gamma_{k-\tau(\gamma_k)}. \tag{51}$$

Proof. First, by the optimality condition of (50) we have

$$\langle \gamma_k G(\bar{x}_k;\xi_k) + x_k - x_{k-1}\,,\, x_{k-1} - x_k\rangle \ge 0,$$

which by rearranging the equation and using (23) we have

$$\|x_k - x_{k-1}\|^2 \le \langle \gamma_k G(\bar{x}_k;\xi_k)\,,\, x_{k-1} - x_k\rangle \le M\gamma_k\|x_k - x_{k-1}\|,$$

Dividing both sides of the equation above by $x_k - x_{k-1}$ gives

$$\|x_k - x_{k-1}\| \le M\gamma_k. \tag{52}$$

Since $\mu = 0$, $\tilde{x}_{k-1} = x_{k-1}$. Next, we consider

$$\begin{aligned}\langle G(\bar{x}_k;\xi_k) - \nabla f(\bar{x}_k), z - x_{k-1}\rangle &= \langle G(\bar{x}_k;\xi_k) - \nabla f(\bar{x}_k)\,,\, z - x_{k-\tau(\gamma_k)}\rangle \\ &\quad + \langle G(\bar{x}_k;\xi_k) - \nabla f(\bar{x}_k)\,,\, x_{k-\tau(\gamma_k)} - x_{k-1}\rangle.\end{aligned} \tag{53}$$

We now provide upper bounds for each term on the right-hand side of Eq. (53). First, using (23) consider the first term on the right-hand side of Eq. (53)

$$\begin{aligned}&\langle G(\bar{x}_k;\xi_k) - \nabla f(\bar{x}_k), z - x_{k-\tau(\gamma_k)}\rangle \\ &= \langle G(\bar{x}_k;\xi_k) - G(\bar{x}_{k-\tau(\gamma_k)};\xi_k), z - x_{k-\tau(\gamma_k)}\rangle + \langle G(\bar{x}_{k-\tau(\gamma_k)};\xi_k) - \nabla f(\bar{x}_{k-\tau(\gamma_k)}), z - x_{k-\tau(\gamma_k)}\rangle \\ &\quad + \langle \nabla f(\bar{x}_{k-\tau(\gamma_k)}) - \nabla f(\bar{x}_k), z - x_{k-\tau(\gamma_k)}\rangle \\ &\le 2L\|\bar{x}_k - \bar{x}_{k-\tau(\gamma_k)}\|\|z - x_{k-\tau(\gamma_k)}\| + \langle G(\bar{x}_{k-\tau(\gamma_k)};\xi_k) - \nabla f(\bar{x}_{k-\tau(\gamma_k)}), z - x_{k-\tau(\gamma_k)}\rangle \\ &\le 4DL\sum_{t=k-\tau(\gamma_k)}^{k-1}\|\bar{x}_{t+1} - \bar{x}_t\| + \langle G(\bar{x}_{k-\tau(\gamma_k)};\xi_k) - \nabla f(\bar{x}_{k-\tau(\gamma_k)}), z - x_{k-\tau(\gamma_k)}\rangle \\ &\le 8D^2L\tau(\gamma_k)\alpha_{k-\tau(\gamma_k)} + \langle G(\bar{x}_{k-\tau(\gamma_k)};\xi_k) - \nabla f(\bar{x}_{k-\tau(\gamma_k)}), z - x_{k-\tau(\gamma_k)}\rangle,\end{aligned}$$

where $\tau(\gamma_k)$ be the mixing time of the underlying Markov chain associated with the step size γ_k, defined in (3). We denote by $\mathcal{F}_k$ the filtration containing all the history generated by the algorithm up to time k. Then, using (32) we have

$$\begin{aligned}&\mathbb{E}[\langle G(\bar{x}_k;\xi_k) - \nabla f(\bar{x}_k), z - x_{k-\tau(\gamma_k)} \,|\, \mathcal{F}_{k-\tau(\gamma_k)}\rangle] \\ &\le 8D^2L\tau(\gamma_k)\alpha_{k-\tau(\gamma_k)} + \|z - x_{k-\tau(\gamma_k)}\|\|\mathbb{E}[G(\bar{x}_k;\xi_k) - \nabla f(\bar{x}_k) \,|\, \mathcal{F}_{k-\tau(\gamma_k)}\| \\ &\le 8D^2L\tau(\gamma_k)\alpha_{k-\tau(\gamma_k)} + 2D\gamma_k.\end{aligned} \tag{54}$$

Second, using Eqs. (52) and (23) we consider the second term on the right-hand side of (53)

$$\begin{aligned}&\langle G(\bar{x}_k;\xi_k) - \nabla f(\bar{x}_k)\,,\, x_{k-\tau(\gamma_k)} - x_{k-1}\rangle \le 2M\|x_{k-\tau(\gamma_k)} - x_{k-1}\| \\ &\le 2M \sum_{t=k-\tau(\gamma_k)}^{k-2} \|x_{t+1} - x_t\| \le 2M^2\tau(\gamma_k)\gamma_{k-\tau(\gamma_k)}. \end{aligned} \tag{55}$$

Taking the expectation on both sides of (53) and using Eqs. (54), (55), and $\alpha_k \le \gamma_k$ immediately gives Eq. (51). □

A.2.2 Proof of Theorem 3

Similar to Lemma 3, we start with the following lemma.

Lemma 4. *Let the sequences $\{x_k, y_k\}$ be generated by Algorithm 2 and $\tilde{x}$ is defined in* (43)*. Then*

$$\begin{aligned}\mathbb{E}[\langle G(y_k;\xi_k) - \nabla f(y_k), z - \tilde{x}_{k-1}\rangle] \le (2M^2 + 4\mu MD + 24\mu LD^2)\tau(\gamma_k)\gamma_{k-\tau(\gamma_k)} \\ + (2D + 2M^2 + 8\mu MD)\gamma_k. \end{aligned} \tag{56}$$

Proof. First, by the optimality condition of (21) we have

$$\langle \gamma_k G(y_k;\xi_k) + \mu\gamma_k(x_k - y_k) + x_k - x_{k-1}\,,\, x_{k-1} - x_k\rangle \ge 0,$$

which by rearranging the equation and using (23) we have

$$\|x_k - x_{k-1}\|^2 \le \langle \gamma_k G(y_k;\xi_k) + \mu\gamma_k(x_k - y_k)\,,\, x_{k-1} - x_k\rangle \le (M + 2D\mu)\gamma_k\|x_k - x_{k-1}\|,$$

Dividing both sides of the equation above by $x_k - x_{k-1}$ gives

$$\|x_k - x_{k-1}\| \le (M + 2D\mu)\gamma_k. \tag{57}$$

Second, we consider

$$\begin{aligned}y_{k+1} - y_k &= \bar{x}_k - \bar{x}_{k-1} - \beta_{k+1}(\bar{x}_k - x_k) + \beta_k(\bar{x}_{k-1} - x_{k-1}) \\ &= \alpha_k(x_k - \bar{x}_{k-1}) - -\beta_{k+1}(\bar{x}_k - x_k) + \beta_k(\bar{x}_{k-1} - x_{k-1}),\end{aligned}$$

which implies that

$$\|y_{k+1} - y_k\| \le 2D\alpha_k + 4D\beta_k \le 6\mu D\gamma_k, \tag{58}$$

where we use the fact that $\beta_k \le \alpha_k = \mu\gamma_k$. Third, we consider

$$\begin{aligned}\langle G(y_k;\xi_k) - \nabla f(y_k), z - \tilde{x}_{k-1}\rangle &= \langle G(y_k;\xi_k) - \nabla f(y_k)\,,\, z - x_{k-\tau(\gamma_k)}\rangle \\ &\quad + \langle G(y_k;\xi_k) - \nabla f(y_k)\,,\, x_{k-\tau(\gamma_k)} - x_k\rangle \\ &\quad + \langle G(y_k;\xi_k) - \nabla f(y_k)\,,\, x_k - \tilde{x}_{k-1}\rangle. \end{aligned} \tag{59}$$

Note that by (43) we have

$$x_k - \tilde{x}_{k-1} = x_k - \frac{1}{1+\mu\gamma_k}x_{k-1} - \frac{\mu\gamma_k}{1+\mu\gamma_k}y_k = x_k - x_{k-1} + \frac{\mu\gamma_k}{1+\mu\gamma_k}(x_{k-1} - y_k),$$

which by substituing into Eq. (59) yields

$$\begin{aligned}\langle G(y_k;\xi_k)-\nabla f(y_k), z-\tilde{x}_{k-1}\rangle &= \langle G(y_k;\xi_k)-\nabla f(y_k)\,,\, z-x_{k-\tau(\gamma_k)}\rangle \\ &\quad + \langle G(y_k;\xi_k)-\nabla f(y_k)\,,\, x_{k-\tau(\gamma_k)}-x_k\rangle \\ &\quad + \langle G(y_k;\xi_k)-\nabla f(y_k)\,,\, x_k-x_{k-1}\rangle \\ &\quad + \frac{\mu\gamma_k}{1+\mu\gamma_k}\langle G(y_k;\xi_k)-\nabla f(y_k)\,,\, x_k-y_k\rangle. \end{aligned} \tag{60}$$

Next, we provide upper bounds for each term on the right-hand side of Eq. (60). Using (23) and Assumption 3, consider the first term on

$$\begin{aligned}&\langle G(y_k;\xi_k)-\nabla f(y_k), z-x_{k-\tau(\gamma_k)}\rangle \\ &= \langle G(y_k;\xi_k)-G(y_{k-\tau(\gamma_k)};\xi_k), z-x_{k-\tau(\gamma_k)}\rangle + \langle G(y_{k-\tau(\gamma_k)};\xi_k)-\nabla f(y_{k-\tau(\gamma_k)}), z-x_{k-\tau(\gamma_k)}\rangle \\ &\quad + \langle \nabla f(y_{k-\tau(\gamma_k)})-\nabla f(y_k), z-x_{k-\tau(\gamma_k)}\rangle \\ &\le 4DL\|y_k-y_{k-\tau(\gamma_k)}\| + +\langle G(y_{k-\tau(\gamma_k)};\xi_k)-\nabla f(y_{k-\tau(\gamma_k)}), z-x_{k-\tau(\gamma_k)}\rangle \\ &\le 4DL\sum_{t=k-\tau(\gamma_k)}^{k-1}\|y_{t+1}-y_t\| + \langle G(y_{k-\tau(\gamma_k)};\xi_k)-\nabla f(y_{k-\tau(\gamma_k)}), z-x_{k-\tau(\gamma_k)}\rangle \\ &\le 24\mu LD^2\sum_{t=k-\tau(\gamma_k)}^{k-1}\gamma_t + \langle G(y_{k-\tau(\gamma_k)};\xi_k)-\nabla f(y_{k-\tau(\gamma_k)}), z-x_{k-\tau(\gamma_k)}\rangle \\ &\le 24\mu LD^2\tau(\gamma_k)\gamma_{k-\tau(\gamma_k)} + +\langle G(y_{k-\tau(\gamma_k)};\xi_k)-\nabla f(y_{k-\tau(\gamma_k)}), z-x_{k-\tau(\gamma_k)}\rangle, \end{aligned}$$

where the second last inequality is due to (58). Taking the conditional expectation w.r.t $\mathcal{F}_k$ and using (32) we have

$$\mathbb{E}[\langle G(y_k;\xi_k)-\nabla f(y_k), z-x_{k-\tau(\gamma_k)}\rangle \,|\, \mathcal{F}_{k-\tau(\gamma_k)}\rangle] \le 24\mu LD^2\tau(\gamma_k)\gamma_{k-\tau(\gamma_k)} + 2D\gamma_k. \tag{61}$$

Second, using Eqs. (57) and (23) we consider the second and third terms on the right-hand side of (60)

$$\begin{aligned}&\langle G(y_k;\xi_k)-\nabla f(y_k)\,,\, x_{k-\tau(\gamma_k)}-x_k\rangle + \langle G(y_k;\xi_k)-\nabla f(y_k)\,,\, x_k-x_{k-1}\rangle \\ &\le 2M\|x_{k-\tau(\gamma_k)}-x_k\| + 2M\|x_k-x_{k-1}\| \le 2M\sum_{t=k+1-\tau(\gamma_k)}^{k}\|x_t-x_{t-1}\| + 2M\|x_k-x_{k-1}\| \\ &\overset{(57)}{\le} 2M(M+2\mu D)\sum_{t=k+1-\tau(\gamma_k)}^{k}\gamma_t + 2M(M+2\mu D)\gamma_k \le 2M(M+2\mu D)\big[\tau(\gamma_k)\gamma_{k-\tau(\gamma_k)}+\gamma_k\big], \end{aligned} \tag{62}$$

where the last inequality is due to the fact that γ_k is nonincreasing. Finally, using (23) we consider the last term of Eq. (60)

$$\frac{\mu\gamma_k}{1+\mu\gamma_k}\langle G(y_k;\xi_k)-\nabla f(y_k)\,,\, x_k-y_k\rangle \le \frac{4\mu MD\gamma_k}{1+\mu\gamma_k}. \tag{63}$$

Taking the expectation on both sides of (60) and using Eqs. (61)–(63) immediately gives Eq. (56). □

Proof of Theorem 3. First, using (10) and (28) gives $\Gamma_0 = 1$, $\alpha_0 = 2$, and $\gamma_0 = 2/\mu$. Second, it is straightforward to verify that (28) satisfies (24) and (38). Thus, using (56) into (39) and since $L = 0$ we

have

$$f(\bar{x}_k) - f(x^*) \leq [f(\bar{x}_0) + 3\mu D]\Gamma_k + \Gamma_k \sum_{t=1}^{k} \frac{4M^2\gamma_t\alpha_t}{\Gamma_t(1+\mu\gamma_t)} + (2M^2 + 4\mu MD + 24\mu LD^2)\Gamma_k \sum_{t=1}^{k} \frac{\tau(\gamma_t)\alpha_t\gamma_{t-\tau(\gamma_t)}}{\Gamma_t} + 2(D + M^2 + 4\mu MD)\Gamma_k \sum_{t=1}^{k} \frac{\alpha_t\gamma_t}{\Gamma_t}. \tag{64}$$

Next, consider each summand on the right-hand side of (64). Using (28) and (10) (to have $\Gamma_t = 2/t(t+1)$) yields

$$\sum_{t=1}^{k} \frac{\gamma_t\alpha_t}{\Gamma_t(1+\mu\gamma_t)} = \sum_{t=1}^{k} \frac{4t(t+1)}{2\mu(t+1)^2(1+\frac{2}{t+1})} = \sum_{t=1}^{k} \frac{2t}{\mu(t+3)} \leq \frac{2k}{\mu}. \tag{65}$$

Using (3) and (28) gives $\tau(\gamma_k) = \log(\mu(k+1)/2)$. Theregore, $\mu(k+1)/2 \geq \tau(\gamma_k)$ and we obtain

$$\gamma_{k-\gamma(\gamma_k)} = \frac{2}{\mu(k+1-\log(\mu(k+1)/2))} \leq \frac{(2+\mu)\tau(\gamma_t)}{\mu k}.$$

Using the relation above, (25), and $\Gamma_t = 2/(t(t+1))$ gives

$$\sum_{t=1}^{k} \frac{\alpha_t\tau(\gamma_t)\gamma_{t-\tau(\gamma_t)}}{\Gamma_t} \leq \frac{2+\mu}{\mu} \sum_{t=1}^{k} \tau(\gamma_t) \leq \frac{(2+\mu)(k+1)\log(\mu(k+1)/2)}{\mu}. \tag{66}$$

Finally, we consider

$$\sum_{t=1}^{k} \frac{\alpha_t\gamma_t}{\Gamma_t} \leq \frac{2k}{\mu}. \tag{67}$$

Using (65)–(67) into (64) together with $\Gamma_k = 2/k(k+1)$ immediately gives (30), i.e.,

$$\begin{aligned} f(\bar{x}_k) - f(x^*) &\leq \frac{2f(\bar{x}_0) + 6\mu D}{k(k+1)} + \frac{8M^2}{\mu(k+1)} + \frac{2(D + M^2 + 4\mu MD)}{\mu(k+1)} \\ &\quad + \frac{4(M^2 + 2\mu MD + 12\mu LD^2)(2+\mu)[\log(\mu/2) + \log(k+1)]}{\mu k} \\ &= \frac{2f(\bar{x}_0) + 6\mu D}{k(k+1)} + \frac{2D + 10M^2 + 8\mu MD}{\mu(k+1)} \\ &\quad + \frac{4(M^2 + 2\mu MD + 12\mu LD^2)(2+\mu)\log(\frac{\mu(k+1)}{2})}{\mu k}. \end{aligned}$$

□

B More details of environments in Section 5

B.1 GridWorld Simulations

We use discount factor of 0.9 in all GridWorld environments. We use the episodic version of TD(0) where we set the episode length to be 10 times the grid size, i.e. episode length is 100 for a 10×10 grid. The learning rate of TD(0) is set to 0.001 while we follow (28) to set the stepsizes β_k and α_k for TD(0)-Acc while we use $\gamma_k = \delta \frac{2}{\mu(k+1)}$ with $\delta = 0.1$. Due to the episodic nature of the problems, these stepsizes are adapted at the episodic level.

B.2 REINFORCE Simulations

We present all parameters setup for all environments with REINFORCE variants in Table 1. The network parameters are specified as $(\text{input} \times \text{hidden layers} \times \text{output})$, i.e. a $4 \times 8 \times 2$ network contains 1 hidden layer of 8 neurons.

Table 1: Parameters for `Acrobot-v1`, `CartPole-v0`, `Swimmer-v2`, `Walker2d-v2`, `HalfCheetah-v2`, and `Ant-v2` environments.

Environment	Algorithm	Policy Network	Discount Factor	Episode Length	Baseline	Batch Size	Learning Rate
CartPole-v0	REINFORCE	4x8x2	0.99	200	None	25	0.1
	REINFORCE-Acc					25	0.1
Acrobot-v1	REINFORCE	6x16x3	0.99	500	None	25	0.1
	REINFORCE-Acc					25	0.1
Swimmer-v2	REINFORCE	8x32x32x2	0.99	1000	Linear	100	0.01
	REINFORCE-Acc					100	0.01
Walker2d-v2	REINFORCE	17x32x32x6	0.99	1000	Linear	100	0.05
	REINFORCE-Acc					100	0.05
HalfCheetah-v2	REINFORCE	17x32x32x6	0.99	1000	Linear	100	0.05
	REINFORCE-Acc					100	0.05
Ant-v2	REINFORCE	111x128x64x32x8	0.99	1000	Linear	100	0.01
	REINFORCE-Acc					100	0.01

Table 2 specifies the descriptions of 6 environments used in this paper including the observation and action spaces.

Table 2: Descriptions of environments used for numerical experiments.

Environment	Observation Space	Action Space	Action Type	Descriptions
Acrobot-v1	2	3	Discrete	The Acrobot-v1 environment contains two joints and two links where we can actuate the joints between two links. The links are hanging downwards at the beginning and the goal is to swing the end of the lower link up to a given height.
CartPole-v0	4	2	Discrete	A pole is attached by an un-actuated joint to a cart moving along a frictionless track. The pole starts upright, and the goal is to prevent it from falling over.
Swimmer-v2	8	2	Continuous	The goal is to make a four-legged creature walk forward as fast as possible.
Walker2d-v2	8	2	Continuous	The goal is to make a two-dimensional bipedal robot walk forward as fast as possible.
HalfCheetah-v2	17	6	Continuous	Make a two-legged creature move forward as fast as possible.
Ant-v2	111	8	Continuous	Make a four-legged creature walk forward as fast as possible.