An Out-of-Shelf Multi-Level Monte Carlo Approach for Average-Reward Reinforcement Learning

Alexander Chernyavskiy
Moscow Institute of Physics and Technology
chernyavskij.as@phyestech.edu

Andrey Veprikov
Moscow Institute of Physics and Technology
Ivannikov Institute for System Programming

Vladimir Solodkin
Moscow Institute of Physics and Technology
Ivannikov Institute for System Programming

Aleksandr Beznosikov
Moscow Institute of Physics and Technology
Ivannikov Institute for System Programming
Innopolis University

Aleksandr Panov
Moscow Institute of Physics and Technology
Federal Research Center "Computer Science and Control"

Abstract

Most modern stochastic optimization methods assume that the data samples are independently identically distributed. However, this assumption is often violated for reinforcement learning setup that deals with temporal-dependent data, coming from a Markov decision process (MDP). Furthermore, to learn reinforcement learning policies, the algorithms have to possess some knowledge about MDP's mixing time or its asymptotic behaviour. For MDPs with high-dimensional state spaces or ones with sparse rewards, mixing time could not be exactly estimated or even may be unknown, making most methods inapplicable. Fortunately, multi-level Monte Carlo approach, taking into account the nature of Markov Chains and letting control variance of the updates, have recently been popularised in the field. The employment of these technique enables the design of reinforcement learning algorithms that are not reliant on oracle knowledge of the mixing time or any assumptions regarding the rate of decay. In light of the aforementioned considerations, we propose an algorithm called MAdam, extending classical Adam for average-reward reinforcement learning. The method leverages non-convex optimization and does not require knowledge of the mixing time. We also provide the theoretical analysis of the optimization procedure and conduct experiments on challenging environments, indicating the qualitative performance of our approach.

1 Introduction

Stochastic gradient methods have always been a key component to solve various optimization problems in deep reinforcement learning (RL) (Schulman et al., 2017; Sutton & Barto, 2018; Hessel et al., 2018). However, most modern optimization methods expect data samples to be independently and identically distributed (i.i.d.). In reinforcement learning problems, temporal dependence of data, caused by Markov property, breaks the assumption which makes theoretical analysis of RL algorithms much more challenging.

Under Markovian setting, there are convergence analyzes (for example, Qiu et al. (2021)) of iterative methods for RL. Typically, consider the rate at which MDP's transition dynamics converge to its stationary distribution, implying the fixed optimal policy. One of the most important elements of the theoretical analysis is mixing time of the MDP and restrictions that can be put on it. In the literature, there are two main ways how to handle the restrictions: prior (oracle) knowledge about the mixing time is employed to determine step size selection, or a hypothesis about exponential decay of the mixing time, such that the data is asymptotically i.i.d. Notably, estimation of mixing time for arbitrary MDP can be computationally expensive (Wolfer, 2020). In most of practical RL applications, an environment posses non-linear transition dynamics and often has sparse reward function. That highlights the exploration problem: the agent can explore long enough without getting the reward signal — i.e. mixing time, usually unknown for the environment, decays much slower than exponentially, violating the hypothesis of exponential mixing. Finally, if the environment suggests a multiple reward scenarios or multitask learning, the environment can have non-linear mixing time decay rate.

In the paper, we focus on the continual average-reward reinforcement learning objective. The latter (Mahadevan, 1996) has a lot of applications in robotics and transportation. From the analytical perspective, this setting has explicit notion of the mixing time. There has been done plenty of research on accelerated stochastic gradient methods, using gradient normalisation (Dorfman & Levy, 2022) or various variants of gradient descent for convex and non-convex optimization problems (Vaswani et al., 2019; Beznosikov et al., 2024). Furthermore, there is a recent line of work inspired by (Dorfman & Levy, 2022), leveraging the multi-level Monte-Carlo method (MLMC) to develop a gradient estimator, aware of the Markovian nature of the incoming data, assuming that the underlying Markov chain is uniformly geometrically ergodic developing a method, converging as square root or the mixing time. Furthermore, there were developed methods with improved convergence rates (Beznosikov et al., 2024), utilisation randomisation of the batch size to improve convergence guarantees (from the perspective of calls to the oracle) for convex and non-convex problems.

Another noteworthy approach that we employ to enhance the algorithmic performance is the Multi-Level Monte Carlo . In reinforcement learning, data is usually represented as a roll-out (sequence of samples from an environment), and during the optimization procedure, the gradients are computed with respect to this roll-out . Consequently, different algorithms may require different number of samples N_t from the MDP to perform the optimization step. For instance, methods that assume data samples are i.i.d., use $N_t = 1$ samples per iteration. However, due to a high dimensionality of the state space, or comprehensive transition dynamics, the cost of getting samples from the environment and thus, getting individual gradients, increases as the distribution of the underlying Markov chain progresses towards a stationary. In order to handle this problem, the MLMC (Giles, 2015) approach was developed. Its main idea is to reduce the computational cost of getting samples by performing most of the simulations at the low cost (in this case, without computing gradients) and getting high cost simulations for very few data samples.

We aim to apply the accelerated gradient methods with extensions of MLMC and with a randomised batch size (roll-out length of concurrent environment samples) for a problem of continual reinforcement learning, targetting mirror policy optimization (Tomar et al., 2020) for convergence analysis. We summarise our contribution as follows:

1. We extend state-of-the-art stochastic optimization method Adam (Kingma & Ba, 2014) with multi-level Monte Carlo gradient estimation for lesser gradient variance and prove its convergence rate. We call the new method `Markovian Adam`, or `MAdam` (see Algorithm 1).
2. We develop an average-reward reinforcement learning algorithm based on mirror descent policy optimization (Tomar et al., 2020) and average-reward policy optimization (Ma et al., 2021) to propose a reinforcement learning algorithm, leveraging convex optimization (see Algorithm 2.
3. We show the practicality of our approach, applying one to a challenging navigation environment with discrete actions, high-dimensional vector observations and sparse reward function.

2 Background

In this section, we are going to introduce the basic reinforcement learning notation (Sutton & Barto, 1998) and connect it with the classical stochastic optimization problem.

2.1 AVERAGE-REWARD REINFORCEMENT LEARNING

Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbb{P}, r)$ be a Markov Decision Process (MDP), where $\mathcal{S}$ is a finite state space, $\mathcal{A}$ is a finite action space, $\mathbb{P}(\cdot|s,a) : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is a transition function that maps the current state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$ into the probability distribution of the next states, and $r : \mathcal{S} \times \mathcal{A} \to [0,\ r_{max}]$ is a reward function, where r_{max} is a certain positive scalar.

Subsequently, the behavioural dynamics of an agent within an MDP is represented by a policy function, $\pi \in (\Delta(\mathcal{A}))^{\mathcal{S}}$, where $\Delta(\mathcal{A})$ is a probability simplex over $\mathcal{A}$. Denoting the current state and action at the time t by s_t and a_t respectively, we define the value function associated with policy π as follows:

$$V^\pi(s) := \mathbb{E}_{a_t \sim \pi(\cdot|s_t), s_{t+1} \sim \mathbb{P}(\cdot|s_t,a_t)} \left[\sum_{t=0}^{\infty} r(s_t, a_t) | \pi, s_0 = s \right].$$

Given the MDP $\mathcal{M}$ and letting $V^\pi(\mu) := \mathbb{E}_{s\sim\mu}[V^\pi(s)]$, our goal is to find the optimal policy:

$$\pi^* \in \arg\max\nolimits_{\pi \in (\Delta(\mathcal{A}))^{\mathcal{S}}} V^\pi(\mu). \tag{1}$$

Similarly to the value function, the action-value function associated with π is defined by:

$$Q^\pi(s,a) := \mathbb{E}_{a_t \sim \pi(\cdot|s_t), s_{t+1} \sim \mathbb{P}(\cdot|s_t,a_t)} \left[\sum_{t=0}^{\infty} r(s_t, a_t) | \pi, s_0 = s, a_0 = a \right].$$

We also define the difference between the value V and the action-value Q functions as the advantage function $A^\pi(s,a) := Q^\pi(s,a) - V^\pi(s)$.

2.2 STOCHASTIC OPTIMIZATION

The formulation of the optimization problem in the form (1) represents a specific instance of a more general formulation of the stochastic optimization problem:

$$\min_{\theta \in \mathbb{R}^d} \left\{ f(\theta) := \mathbb{E}_{Z\sim\mu} \left[f(\theta, Z) \right] \right\}, \tag{2}$$

where μ is a usually unknown distribution. As mentioned earlier, the majority of modern optimization theory techniques operate under the assumption that the sequence of random variables sampled $\{Z_n\}_{n=1}^{\infty}$ is independent and identically distributed . In our case, however, the nature of randomness is derived from the MDP. Consequently, the sequence of random variables $\{Z_n\}_{n=1}^{\infty}$ is considered to be a realization of a Markov chain with a stationary distribution μ. The classical approach to efficiently solve the problem (2), especially in the context of reinforcement learning, is based on methods that utilize adaptive gradient normalization. For instance, RMSProp , AdaGrad and Adam have been demonstrated to perform well when training Deep Q-Networks . Nevertheless, there is a notable absence of literature examining the theoretical analysis of these methods in the context of Markovian noise. In our work, we focus on one of the most popular out-of-box method for reinforcement learning, Adam. In Section 3.1 we present the new algorithm `Markovian Adam` (Algorithm 1) that can be applied to solve the problem (1) in the case of Markovian noise. The convergence analysis is presented in Theorem 1.

3 MAIN RESULTS

Our approach to solve the problem (1) can now be decomposed into the two key parts: mirror descent policy optimization and policy improvement using Algorithm 1.

3.1 MARKOVIAN ADAM OPTIMIZATION ALGORITHM

Now, we present our algorithm `Markovian Adam` (Algorithm 1) that solves the problem (1) in the Markovian nose setting using the MLMC approach.

In line 11 of Algorithm 1, the index $N(T)$ is distributed according to the rule

$$\forall j \in \mathbb{N} : j < T \hookrightarrow \mathbb{P}\{N(T) = j\} \propto 1 - \beta_1^{T-j}. \tag{3}$$

Algorithm 1 `Markovian Adam (MAdam)`

```
Parameters: step sizes $\{\alpha_t\}_{t=0}^{T} \subset \mathbb{R}_+$, exponential decay rates for the momentum estimates $0 \leq \beta_1 < \beta_2 < 1$, number of iterations $T$, batch size $B$, limit $M$ and noise $\varepsilon$.
Initialization: $\theta^0 \in \mathbb{R}^d$, $m^0 = v^0 = 0 \in \mathbb{R}^d$.
for $t = 0, 1, 2, \ldots, T$ do
    Sample $J_t \sim \mathrm{Geom}(1/2)$
    $g^t = \begin{cases} 2^{J_t}(g^t_{J_t} - g^t_{J_t - 1}), & \text{if } 2^{J_t} \leq M \\ g^t_0, & \text{otherwise} \end{cases}$ with $g^t_j = 2^{-j} B^{-1} \sum_{i=1}^{2^j B} \nabla f(\theta^t, Z_{n^t + i})$
    $m^{t+1} = \beta_1 m^t + g^t$
    $v^{t+1} = \beta_2 v^t + g^t \odot g^t$
    $\theta^{t+1} = \theta^t - \alpha_t m^{t+1} / \sqrt{v^{t+1} + \varepsilon}$
    $n^{t+1} = n^t + 2^{J_t} B$
end for
Output: $\theta^{N(T)}$, where $N(T)$ is distributed according to (3).
```

If $\beta_1 = 0$, then $N(T)$ distributed uniformly, if $\beta_1 \in (0; 1)$, then outcomes from the first iterations of Algorithm 1 will be used with lower probability. We now provide several assumptions, required for the convergence analysis of Algorithm 1.

Assumption 1. The function $f(\theta)$ is L-smooth on $\mathbb{R}^d$, i.e., it is differentiable and there exists $L > 0$ such that for any $\theta, \phi \in \mathbb{R}^d$ the following inequality holds

$$\|\nabla f(\theta) - \nabla f(\phi)\| \leq L \|\theta - \phi\|.$$

Assumption 2. The gradient estimator $\nabla f(\theta, Z)$ is uniformly almost surely bounded, i.e, there exists $R \geq \sqrt{\varepsilon}$ such that for any $\theta \in \mathbb{R}^d$

$$\|\nabla f(\theta, Z)\| \leq R - \sqrt{\varepsilon}.$$

Assumption 3. $\{Z_t\}_{t=0}^{\infty}$ is a stationary Markov chain on $(\mathsf{Z}, \mathcal{Z})$ with unique invariant distribution μ. Moreover, $\{Z_t\}_{t=0}^{\infty}$ is uniformly geometrically ergodic with mixing time τ_{mix}, i.e. for all $t > 0$, $z_0, z \in \mathcal{Z}$ the following inequality holds

$$|\mathbb{P}\{Z_t = z | Z_0 = z_0\} - \mu_z| \lesssim (1/2)^{t/\tau_{\text{mix}}}.$$

Assumptions 2 and 3 are classical in the literature considering Markovian noise (Creswell et al., 2018; Dorfman & Levy, 2022; Beznosikov et al., 2024). Whereas in the case of i.i.d. noise the gradient norm can be bounded in expectation, i.e., $\mathbb{E}_{Z \sim \mu}\left[\|\nabla f(\theta, Z)\|^2\right] \leq R^2$, Assumption 2 bounds the gradient norm uniformly. This complication is associated with the nature of Markovian noise, and, to the best of our knowledge, no existing literature has proposed an alternative approach (Doan et al., 2020; Even, 2023; Solodkin et al., 2024). The $\sqrt{\varepsilon}$ term in Assumption 2 helps to simplify the final bounds.

We now ready to provide the convergence rate of `MAdam` (Algorithm 1).

Theorem 1 (Convergence of `MAdam` (Algorithm 1).)**.** Let Assumptions 1, 2, 3 be satisfied. Given the iterates defined by system (11), $0 \leq \beta_1 < \beta_2 < 1$, $\alpha_t = \alpha(1 - \beta_1)\sqrt{\frac{1 - \beta_2^t}{1 - \beta_2}}$ and $N(T)$ defined by (3). Then for any T such that $T > \max\{\frac{\beta_1}{1 - \beta_1} \ ; \ \tau_{\text{mix}}\}$ it holds that

$$\mathbb{E}\left[\left\|\nabla f(\theta^{N(T)})\right\|^{\frac{4}{3}}\right]^{\frac{3}{2}} = \mathcal{O}\Bigg(R_g \frac{f(\theta^0) - f^*}{\alpha \tilde{T}} + \frac{\Delta}{\tilde{T}}\left[\ln\left(1 + \frac{R_g^2}{\varepsilon(1 - \beta_2)}\right) - T\log(\beta_2)\right]$$
$$+ \frac{d R_g \mu_g}{\tilde{T}}\left[T - \tau_{\text{mix}} + \left(\frac{\beta_1}{\beta_2}\right)^{-\tau_{\text{mix}}} \Delta_{\tau_{\text{mix}}}\right] \frac{1 - \beta_1}{(1 - \beta_1/\beta_2)\sqrt{1 - \beta_2}}\Bigg),$$

where $R_g^2 := (1 + B^{-1}\tau_{\text{mix}} \log M)R^2$, $\mu_g^2 := B^{-1}\tau_{\text{mix}} M^{-1} R^2$, $\tilde{T} := T - \beta_1/(1-\beta_1)$ and

$$\Delta := \frac{\alpha d R_g L(1-\beta_1)}{(1-\beta_1/\beta_2)(1-\beta_2)} + \frac{2\alpha^2 d L^2 \beta_1}{(1-\beta_1/\beta_2)(1-\beta_2)^{3/2}} + \frac{12 d R_g^2 \sqrt{1-\beta_1}}{(1-\beta_1/\beta_2)^{3/2}\sqrt{1-\beta_2}},$$

$$\Delta_{\tau_{\text{mix}}} := \sum_{t=1}^{\tau_{\text{mix}}} \frac{\|\nabla f(\theta^t)\|}{\sqrt{\mathbb{E}_t\left[\|g^t\|^2\right] + \varepsilon}}.$$

Discussion. The convergence rate in Theorem 1 is similar to the i.i.d. case (Défossez et al., 2020), however we obtain terms of the form $\Delta_{\tau_{\text{mix}}}$, which are connected to the Markovian nature of the noise in the problem (1). We also obtain the convergence rate in terms of $\mathbb{E}\left[\left\|\nabla f(\theta^{N(T)})\right\|^{4/3}\right]^{3/2}$ but not in $\mathbb{E}\left[\left\|\nabla f(\theta^{N(T)})\right\|^{2}\right]$ as in the i.i.d case. This is due to the fact that we need to use the Hölder inequality (see B.1) to use results from Lemma 2. We also obtain terms of the form R_g and μ_g because in line 5 of Algorithm 1 we use MLMC gradient estimator g^t. From Lemma 2 it can be shown that the expected number of oracle calls at each iteration of MLMC estimator is equal to $\mathcal{O}(B \log M)$, and $\mu_g^2 \propto (BM)^{-1}$ and $R_g^2 \propto B^{-1} \log M$.

We now provide the corollary of Theorem 1 where we choose specific parameters of the Algorithm 1.

Corollary 1 (Parameters tuning for Algorithm 1)**.** Under the conditions of Theorem 1 choosing parameters of the `MAdam` algorithm (Algorithm 1) as

$$\alpha_t = \mathcal{O}\left(\sqrt{1 - \left(1 - \frac{1}{T}\right)^t}\right) \;;\; \beta_1 = 0 \;;\; \beta_2 = 1 - \frac{1}{T} \;;$$

$$T \gg \max\left\{\frac{\beta_1}{1-\beta_1} \;;\; \tau_{\text{mix}}\right\} \;;\; M = \mathcal{O}(T^2) \;;\; B = \mathcal{O}(1).$$

Then convergence rate of the algorithm could be re-written in the form of

$$\mathbb{E}\left[\left\|\nabla f(\theta^{N(T)})\right\|^{\frac{4}{3}}\right]^{\frac{3}{2}} = \tilde{\mathcal{O}}\left(\frac{\sqrt{\tau_{\text{mix}}}R(f(\theta^0) - f^*)}{\sqrt{T}} + \frac{d}{\sqrt{T}}\left(\tau_{\text{mix}}R^2 + \sqrt{\tau_{\text{mix}}}RL\right)\right).$$

3.2 Mirror descent policy optimization

Our method is based on Mirror Descent Policy Optimization (MDPO, Tomar et al. (2020)), that could be formally described as

$$\theta_{t+1} = \arg\max_\theta \mathbb{E}_{s\sim\mu_{\theta_t}(\cdot)}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[A^{\pi_{\theta_t}}(s,a)] - \left(1 - \frac{t}{T}\right)^{-1} \text{KL}(\pi, \pi_{\theta_t})\right], \tag{4}$$

Despite the fact that this update differs from the one in (1) by a KL-divergence regularisation term, it has been shown in (Huang et al., 2021; Alfano et al., 2024), that the procedure (4) converges to the neighbourhood of the solution π^* of the problem (1). During each step, the method approaches a constrained over policy parameters θ optimization problem. In light of the fact that a single gradient step does not enforce the trust region constraint: $\nabla_\theta \text{KL}(\pi, \pi_{\theta_t})|_{\pi=\pi_{\theta_t}} = 0$, the method should take $m > 1$ optimization steps per an update. The problem setting generalises over state-of-the-art methods like TRPO (Schulman et al., 2015a) and PPO (Schulman et al., 2017). Nonetheless, instead of exactly solving the problem, MDPO approximates the solution, taking multiple steps in the direction of the gradient of its objective function. The update rule is consistent with mirror descent (Beck & Teboulle, 2003) update rule in convex optimization.

Let us briefly describe the method, following the multi-level actor-critic framework (Patel et al.). Pseudocode for the algorithm is listed in Algorithm 2. The method appears as a classical actor-critic, adapted for the average-reward objective (1). As we consider discrete distributions over the probability simplex, for actor the objective can be defined using policy gradient theorem (Sutton et al., 1999). In the literature (Patel et al.; Ganesh & Aggarwal, 2024; Suttle et al., 2023), the actor

Algorithm 2 Average-reward mirror descent optimization, following (Suttle et al., 2023; Ma et al., 2021; Tomar et al., 2020)

```
Parameters: actor step size $\alpha_t$, critic step size $\beta_t$, average reward tracker step size $\gamma_t$, maximum trajectory length $T_{\text{max}}$, number of iterations $T$
Initialization: actor parameters $\theta_0$, critic parameters $\varphi_0$, initial environment state $s_1^{(0)} \sim \rho$
for $t = 0, \dots, T-1$ do
    Initialise an empty trajectory $\mathcal{T}_t$
    for $i = 1, \dots, H$ do                                        ▷ Environment data collection
        Take an action $a_t^i \sim \pi_{\theta_t}(\cdot|s_t^i)$
        Transition to next env. state $s_t^{i+1} \sim \mathbb{P}(\cdot|s_t^i, a_t^i)$
        Receive reward $r_t^i = r(s_t^i, a_t^i)$ from the env.
        Append the sample $\{s_t^i, a_t^i, r_t^i, s_t^{i+1}\}$ to the trajectory $\mathcal{T}_t$
    end for
    Calculate advantage $A^{\omega_t}(s, a)$ from (6)
    Calculate value and gradients of the actor objective (5)
    Calculate value and gradients of the critic objective (8)
    Get MLMC estimations of gradients and parameters updates according to Algorithm 1
    Update policy parameters $\theta_t$ with (7)
    Update critic parameters $\omega_t$ with (10)
    Update reward $\eta_\pi$ and value trackers $b$ according to (9)
end for
```

requires certain conditions to be satisfied. However, as we allow arbitrary policy parametrisations, we don't make any specific assumptions on the policy other than being differentiable. These conditions are sufficient, when utilising neural networks with general parametrisation and continuous activation function. The gradient of the objective function is defined in the following way:

$$\nabla_\theta J_{\pi_\theta} = \mathbb{E}_{s\sim\mu(\cdot),\; a\sim\pi_\theta(\cdot|s)}[A^{\pi_\theta}(s,a)\log\pi_\theta(a|s)] \tag{5}$$

For advantage $A^{\pi_{\theta_k}}(s,a)$ estimation, we use generalised advantage estimation (GAE, Schulman et al. (2015b)) :

$$A^{\pi_{\theta_k}}(s_t,a_t) = \sum_{l=0}^{\infty}\lambda^l\delta_{t+1} = \sum_{l=0}^{\infty}\lambda^l\left(r_{t+l} - \eta_{\pi_{\theta_k}} + V^{\pi_\theta}(s_{t+l+1}) - V^{\pi_\theta}(s_{t+l})\right) \tag{6}$$

Reasoning of using this notion for advantage is its flexibility. In particular, we consider two cases:

1. $\lambda = 0$ implies that $A^{\pi_{\theta_k}}(s_t,a_t) = r_t - \eta_{\pi_{\theta_k}} + V(s_{t+1}) - V(s_t)$ — $TD(0)$-learning
2. $\lambda = 1$ implies that $A^{\pi_{\theta_k}}(s_t,a_t) = \sum_{l=0}^{\infty}(r_{t+l} - \eta_{\pi_{\theta_k}}) - V(s_t)$ — $TD(\infty)$-learning, or Monte Carlo estimations,

and $0 < \lambda < 1$ establishes balance between bias and variance of the estimations.

Actor's parameters are updated in the following manner with a learning rate α:

$$\theta = \theta + \alpha\nabla_\theta J_{\pi_\theta} \tag{7}$$

For critic, there is a popular assumption in the literature to consider a linear critic with a weight vector ω and feature map $\varphi(s) : ||\varphi(s)|| \leq 1\ \forall s \in \mathcal{S}$, such that $V^\omega(s) = \langle\varphi(s), \omega\rangle$. Critic's objective could be denoted as follows:

$$\begin{aligned} &\min_{\omega\in\Omega}\mathbb{E}_{s\sim\mu(\cdot)}\left[V^{\pi_\theta}(s) - V^\omega(s)\right], \\ &\text{s.t. } \mathbb{E}_{s\sim\mu(\cdot)}[V^\omega(s)] = 0 \end{aligned} \tag{8}$$

Notably, it can be noticed that with this critic approximation there can be "value drift" effect (Ma et al., 2021) — i.e. accumulation of value function error during training. To mitigate the issue, the authors propose to regularise the objective, centring the value function around zero. The problem is equivalent to minimisation of the unconstrained objective with $\hat{V}^{\omega}(s) = V^{\omega}(s) - \nu b$, where b is the average value.

Average value b and reward η_π are updated as follows with learning rate γ:

$$\begin{aligned} \eta_\pi &\leftarrow (1-\gamma)\eta_\pi + \gamma \frac{1}{N}\sum_{t=1}^{N} r(s_t, a_t) \\ b &\leftarrow (1-\gamma)b + \gamma \frac{1}{N}\sum_{t=1}^{N} V(s_t) \end{aligned} \tag{9}$$

Critic updates are done in the way with a learning rate β:

$$\omega \leftarrow \Pi_\Omega \left[\omega - \beta \sum_{l=0}^{\infty} \lambda^l \left(r_{t+l} - \eta_{\pi_\theta} + \langle \omega, s_{t+l+1} \rangle - \langle \omega, s_{t+l} \rangle \right) \varphi(s_{t+l}) \right] \tag{10}$$

where Π_Ω projection (i.e. softmax) projection operator to the critic's parameter space Ω.

4 Experiments

We conduct experiments, utilising a set of challenging maps from grid-based navigation suite Pignatelli et al. (2024) to investigate, how the method behaves in case of non-linear mixing time dependence caused by necessity of the environment exploration. All the experiments were reported as an average of five random seeds, as well as the 95 % confidence interval. Our goal is not to demonstrate the state-of-the-art performance of the optimisation method but provide a proof of concept that using `MAdam` with its MLMC gradient estimator can help build efficient algorithms, outperforming a baseline, utilising Adam as out-of-shelf widely known approach. Thus, we don't specifically tune any hyperparameters of the algorithm.

`MAdam` experimental setup. We take a basic Adam (Kingma & Ba, 2014) optimiser from `optax`[1] library with hyperparameters specified by the corollary 1 and exponential learning rate schedule and equip it with the MLMC gradient estimator (Algorithm 2, line 5). It was noticed (Dorfman & Levy, 2022) that using standard geometric distribution is not viable due to the presence of high-variant gradients, slowing down the training process, so we draw J from truncated geometric distribution $P(J=j) \sim 2^{-j}, j \in \{1, \dots, K\}$, where $K = 5$ is a fixed parameter. As large values of J are low probable, we didn't tune K. Furthermore, we set batch size bound parameter B to overall generated environment roll-out length and batch size limit $M = 32$ as a maximum J that can be drawn from the truncated geometric distribution. Overall, we compare our method with popular optimisers like the vanilla Adam, AdamW and stochastic gradient descent with momentum (Nesterov, 1983).

Reinforcement algorithm setup. As the algorithm has the randomised batch size bound, or randomised roll-out length parameter B, it does make sense to set the computational budget not in a number of training iterations but in a total number of observed environment samples during the training. We set this number to $T = 10^6$ as the budget, necessary to reach the MDP's stationary distribution and stabilise the average reward η_π. As for the advantage estimation (6), we limit our research to a specific case of the advantage estimation (Ma et al. (2021), $\lambda = 0.95$) and leave the general setting for the future work. When constructing a batch of environment trajectories of 128, we use vectorisation of the environments to keep large batches of 4 and not making the trajectories too long. For the experiments, we use the same learning rate equal to $2.5 \cdot 10^{-4}$ for all the actor-critic components.

We chose two simple environments[2], targetting different objects of exploration and non-linear mixing time dependency (Figure 1):

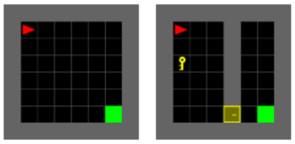

Figure 1: Navigation environment, used in the paper: empty goal reaching environment (left) and an environment with an additional "key-door" subgoal (right).

[1] `https://optax.readthedocs.io/en/latest/`

[2] `https://github.com/epignatelli/navix`

1. Empty 5×5 goal reaching task, where an agent has to move from the upper left corner of the grid to the lower right, being rewarded only upon reaching the goal — with difficulty of exploration it could be sophisticated to establish the stationary distribution.
2. 5×5 goal reaching task with a door, where there is an additional subgoal to reach before the original environment goal.

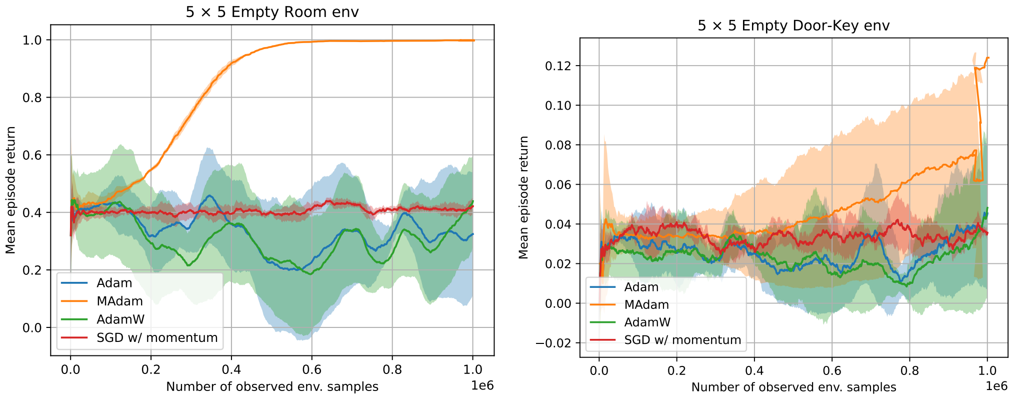

Figure 2: Results of the experiments with the empty 5×5 room environment (left) and 5×5 "key with door" environment (right). The method, using `MAdam`, outperforms the baselines on both environments.

For the empty room environment depicted in the left part of figure 2, RL algorithm, using MLMC reaches the perfect mean episode return of 1.0, whereas the baseline method succeeds in twice fewer cases. In the environment with the subgoal (figure 2, right), both methods struggle because of necessity of exploration and existence of more complicated memory structure than a scalar average reward tracker, however, due to gradient estimation with longer roll-outs, the method, employing `MAdam` reaches higher mean episode returns than the baseline.

Additionally, we conduct experiments on a robotic navigation environment[3] with high-dimensional observations, simulating a LiDAR sensor's signals from a $360°$ arc centred on the robot's forward axis. The environment implements the goal reaching task, with goals randomly sampled on the map. For a scheme of the environment and results, see Figure 3.

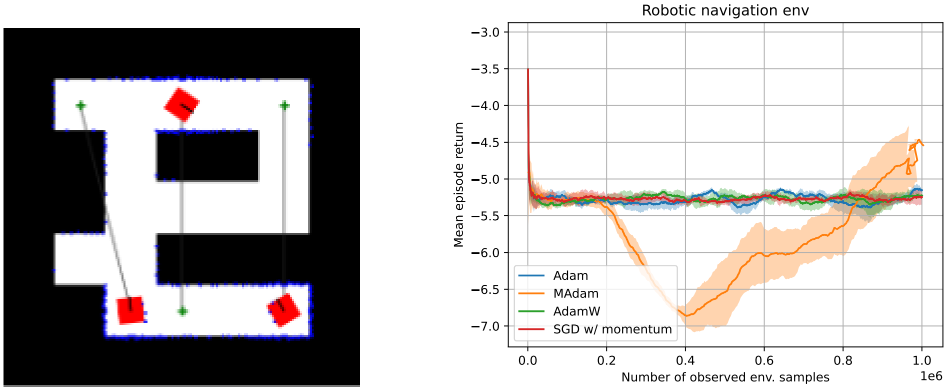

Figure 3: The graphical description of the environment from (Rutherford et al., 2024) (left), we use a single-agent version and experimental results (right). The method, using `MAdam`, outperforms the baselines on both environments despite initial slump caused by the exploration.

[3] `https://github.com/FLAIROx/JaxMARL/tree/main/jaxmarl/environments/jaxnav`

5 Conclusion

In this paper, we studied how an out-of-shelf optimization approach for average-reward reinforcement learning could be improved with multi-level Monte Carlo gradient estimates, one we called `MAdam`. We proved a key theoretical result about the method convergence (Theorem 1) and choice of the optimiser's hyperparameters (Corollary 1). Furthermore, we developed an average-reward policy optimisation method based on mirror descent policy optimisation to maintain convexity of the problem. We demonstrated the practicality of the method on several navigation environments, having mixing time dependence different from exponential, and showed that the approach with multi-level gradient estimation outperforms the baselines. We hope to continue our work in a direction of setting convergence guarantees for average-reward reinforcement learning methods with general parametrisation.

References

Carlo Alfano, Rui Yuan, and Patrick Rebeschini. A novel framework for policy mirror descent with general parameterization and linear convergence. *Advances in Neural Information Processing Systems*, 36, 2024.

Qinbo Bai, Washim Uddin Mondal, and Vaneet Aggarwal. Regret analysis of policy gradient algorithm for infinite horizon average reward markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 10980–10988, 2024.

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

Aleksandr Beznosikov, Sergey Samsonov, Marina Sheshukova, Alexander Gasnikov, Alexey Naumov, and Eric Moulines. First order methods with markovian noise: from acceleration to variational inequalities. *Advances in Neural Information Processing Systems*, 36, 2024.

Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35 (1):53–65, 2018.

Alexandre Défossez, Léon Bottou, Francis Bach, and Nicolas Usunier. A simple convergence proof of adam and adagrad. *arXiv preprint arXiv:2003.02395*, 2020.

Thinh T. Doan, Lam M. Nguyen, Nhan H. Pham, and Justin Romberg. Finite-time analysis of stochastic gradient descent under markov randomness, 2020.

Ron Dorfman and Kfir Yehuda Levy. Adapting to mixing time in stochastic optimization with markovian data. In *International Conference on Machine Learning*, pp. 5429–5446. PMLR, 2022.

Mathieu Even. Stochastic gradient descent under markovian sampling schemes. In *International Conference on Machine Learning*, pp. 9412–9439. PMLR, 2023.

Swetha Ganesh and Vaneet Aggarwal. An accelerated multi-level monte carlo approach for average reward reinforcement learning with general policy parametrization. *arXiv preprint arXiv:2407.18878*, 2024.

Michael B Giles. Multilevel monte carlo methods. *Acta numerica*, 24:259–328, 2015.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Feihu Huang, Shangqian Gao, and Heng Huang. Bregman gradient policy optimization. *arXiv preprint arXiv:2106.12112*, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Xiaoteng Ma, Xiaohang Tang, Li Xia, Jun Yang, and Qianchuan Zhao. Average-reward reinforcement learning with trust region methods. *arXiv preprint arXiv:2106.03442*, 2021.

Sridhar Mahadevan. Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine learning*, 22(1):159–195, 1996.

Yurii Nesterov. A method for solving the convex programming problem with convergence rate o (1/k2). In *Dokl akad nauk Sssr*, volume 269, pp. 543, 1983.

Bhrij Patel, Wesley A Suttle, Alec Koppel, Vaneet Aggarwal, Brian M Sadler, Dinesh Manocha, and Amrit Bedi. Towards global optimality for practical average reward reinforcement learning without mixing time oracles. In *Forty-first International Conference on Machine Learning*.

Eduardo Pignatelli, Jarek Liesen, Robert Tjarko Lange, Chris Lu, Pablo Samuel Castro, and Laura Toni. Navix: Scaling minigrid environments with jax. *arXiv preprint arXiv:2407.19396*, 2024.

Shuang Qiu, Zhuoran Yang, Jieping Ye, and Zhaoran Wang. On finite-time convergence of actor-critic algorithm. *IEEE Journal on Selected Areas in Information Theory*, 2(2):652–664, 2021.

Alexander Rutherford, Michael Beukman, Timon Willi, Bruno Lacerda, Nick Hawes, and Jakob Foerster. No regrets: Investigating and improving regret approximations for curriculum discovery. *arXiv preprint arXiv:2408.15099*, 2024.

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

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

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

Vladimir Solodkin, Andrew Veprikov, and Aleksandr Beznosikov. Methods for optimization problems with markovian stochasticity and non-euclidean geometry, 2024. URL `https://arxiv.org/abs/2408.01848`.

Wesley A Suttle, Amrit Bedi, Bhrij Patel, Brian M Sadler, Alec Koppel, and Dinesh Manocha. Beyond exponentially fast mixing in average-reward reinforcement learning via multi-level monte carlo actor-critic. In *International Conference on Machine Learning*, pp. 33240–33267. PMLR, 2023.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

R.S. Sutton and A.G. Barto. Reinforcement learning: An introduction. *IEEE Transactions on Neural Networks*, 9(5):1054–1054, 1998. doi: 10.1109/TNN.1998.712192.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. *arXiv preprint arXiv:2005.09814*, 2020.

Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of sgd for over-parameterized models and an accelerated perceptron. In *The 22nd international conference on artificial intelligence and statistics*, pp. 1195–1204. PMLR, 2019.

Geoffrey Wolfer. Mixing time estimation in ergodic markov chains from a single trajectory with contraction methods. In *Algorithmic Learning Theory*, pp. 890–905. PMLR, 2020.

Supplementary Material

A Related work

Stochastic optimisation with Markovian noise. A lot of recent work in optimisation is still focused on i.i.d. setting because of its numerous practical applications in supervised learning, only some works sparsely tackle the problem. With diversity of the optimisation algorithms, many of them can be equipped with Markovian gradient estimators. For instance, RASGD (Beznosikov et al., 2024) considers stochastic gradient descent with different sort of momentum, targetting faster convergence in terms of taking less oracle calls as well as MLMC with randomised batch size; for adaptive gradient methods, MAG (Dorfman & Levy, 2022) has to be highlighted that uses MLMC coupled with adaptive gradient optimisation. Both methods showed convergence guarantees for convex, non-convex problems and variational inequalities. Moreover, both methods have potential to be applied to reinforcement learning, to one-step $TD(0)$ learning. We aim to show that the same group of methods could be relevant to Adam optimiser.

Multi-level Monte-Carlo methods for RL. MLMC has been initially applied to TD-learning (Dorfman & Levy, 2022), however, it has found many more applications in the actor-critic domain for average-reward reinforcement learning. Nonetheless, the first algorithms were computationally intractable and were developed only for linear parametrisations, PPGAE(Bai et al., 2024) was one of the first algorithms that had been designed to work with general parametrisations. Unfortunately, one requires explicit knowledge of the mixing time to compose samples in a set of independent or rather non-overlapping trajectories, that limits its applicability due to computational complexity. To alleviate dependency of the mixing time, there was a developed MAC (Suttle et al., 2023), an $TD(0)$ AC with multi-level Monte-Carlo estimation for actor, critic, and average reward tracker η_π. The algorithm does not require any assumptions on the mixing or hitting time and moreover, has proven global optimality with the square root mixing time dependence and fast convergence (Patel et al.). Furthermore, for general actor-parametrisations there was developed an optimal algorithm (Ganesh & Aggarwal, 2024), employing average-reward reinforcement learning paired with RASGD optimiser. However, the most reasonable and widely used modern reinforcement methods are based on trust region methods, like PPO (Schulman et al., 2017; Ma et al., 2021), so our goal is to demonstrate the applicability of the out-of-box method, minimally adapted to the problem statement.

B Auxiliary Lemmas and Facts

B.1 Hölder inequality

For the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ let $\mathbb{E}[\cdot]$ denote the expectation operator. For real- or complex-valued random variables X, Y on Ω and for all $p, q \geq 0$ such that $p^{-1} + q^{-1} = 1$ it holds that

$$\mathbb{E}[|XY|] \leq \left(\mathbb{E}[|X|^p]\right)^{1/p} \left(\mathbb{E}[|Y|^q]\right)^{1/q}.$$

B.2 Fenchel-Young inequality

For all $x, y \in \mathbb{R}^d$ and $\kappa > 0$ it holds that

$$2\langle x, y\rangle \leq \kappa^{-1}\|x\|^2 + \kappa\|y\|_*^2.$$

C Proof of the convergence rate of the Markovian Adam

In this paper, we build on the proof from the Défossez et al. (2020).

C.1 Notations

In the rest of this section, we take an iteration $t \in \mathbb{N}$, and when needed, $i \in [d]$ refers to a specific coordinate. Given $x^0 \in \mathbb{R}^d$ our starting point, $m^0 = 0$, and $v^0 = 0$, we define

$$\begin{cases} m_i^t &= \beta_1 m_i^{t-1} + g_i^t, \\ v_i^t &= \beta_2 v_i^{t-1} + \left(g_i^t\right)^2, \\ x_i^t &= x_i^{t-1} - \alpha_t m_i^t / \sqrt{\varepsilon + v_i^t}. \end{cases} \tag{11}$$

For Adam, the step size is given by

$$\alpha_t = \alpha(1-\beta_1)\sqrt{\frac{1-\beta_2^t}{1-\beta_2}}. \tag{12}$$

Therefore, this step size is monotonic, i.e. $\alpha_t \geq \alpha_{t-1}$. Throughout the proof we note $\mathbb{E}_t\left[\cdot\right]$ the conditional expectation with respect to $x^0, \dots, x^{t-1}$. In particular, m^{t-1}, v^{t-1} is deterministic knowing $x^0, \dots, x^{t-1}$. We introduce

$$G^t = \nabla f(x^{t-1}) \tag{13}$$

We introduce the update $u^t \in \mathbb{R}^d$ and $U^t \in \mathbb{R}^d$ as

$$u_i^t = \frac{m_i^t}{\sqrt{v_i^t + \varepsilon}} \quad \text{and} \quad U_i^t = \frac{g_i^t}{\sqrt{v_i^t + \varepsilon}}. \tag{14}$$

For any $t \in \mathbb{N}$ with $t < T$, we define $\tilde{v}^{t,k} \in \mathbb{R}^d$ by

$$\tilde{v}_i^{t,k} = \beta_2^k v_i^{t-k} + \mathbb{E}_{t-k-1}\left[\sum_{j=n-k+1}^{t} \beta_2^{t-j}(g_i^j)^2\right], \tag{15}$$

C.2 Technical lemmas

Lemma 1. Let Assumptions 2, 3 be satisfied. Then for any $x \in \mathbb{R}^d$ and $B \geq \tau_{\text{mix}}$ it holds that

$$\mathbb{E}\|B^{-1}\textstyle\sum_{i=1}^{B} \nabla f(x, Z_i) - \nabla f(x)\|^2 \lesssim \frac{\tau_{\text{mix}}}{n}(R - \sqrt{\varepsilon})^2$$

Proof. The proof is to apply Lemma 1 from Solodkin et al. (2024) with $\xi_t = \|B^{-1}\sum_{i=1}^{B} \nabla f(x, Z_i) - \nabla f(x)\|^2,\ \sigma = 2(R - \sqrt{\varepsilon}),\ N = B$ and $\hat{V}(0, x) = \frac{1}{2}\|x\|_2$. □

Lemma 2. Let Assumptions 2, 3 be satisfied. Then for the gradient estimates defined in line 5 of Algorithm 1, it holds that $\mathbb{E}_t[g^t] = \mathbb{E}_t[g^t_{\lfloor \log M \rfloor}]$. Moreover,

$$\begin{aligned} &\mathbb{E}_t\|g^t\|^2 \leq 2(1 + 176B^{-1}\tau_{\text{mix}} \log_2 M)(R - \sqrt{\varepsilon})^2 =: R_g^2, \\ &\|\nabla f(x) - \mathbb{E}_t[g^t]\|^2 \leq 88B^{-1}\tau_{\text{mix}} M^{-1}(R - \sqrt{\varepsilon})^2 := \mu_g^2. \end{aligned}$$

Proof. We first proof that $\mathbb{E}_t[g^t] = \mathbb{E}_t[g^t_{\lfloor \log_2 M \rfloor}]$. Let us enroll the conditional expectation w.r.t. J_t:

$$\begin{aligned} \mathbb{E}_t[g^t] = \mathbb{E}_k\left[\mathbb{E}_{J_t}[g^t]\right] &= \mathbb{E}_t[g_0^k] + \sum_{i=1}^{\lfloor \log_2 M \rfloor} \mathbb{P}\{J_t = i\} \cdot 2^i \mathbb{E}_t[g_i^t - g_{i-1}^t] \\ &= \mathbb{E}_t[g_0^t] + \sum_{i=1}^{\lfloor \log_2 M \rfloor} \mathbb{E}_t[g_i^t - g_{i-1}^t] = \mathbb{E}_t[g^t_{\lfloor \log_2 M \rfloor}]\,. \end{aligned}$$

To proof the first inequality of Lemma 2, we again take the conditional expectation for J_t:

$$\begin{aligned}\mathbb{E}_t[\|\nabla f(x) - g^t\|^2] &\le 2\mathbb{E}_t[\|\nabla f(x) - g_0^t\|^2] + 2\mathbb{E}_t[\|g^t - g_0^t\|^2] \\ &= 2\mathbb{E}_t[\|\nabla f(x) - g_0^t\|^2] \\ &\quad + 2\sum\nolimits_{i=1}^{\lfloor \log_2 M \rfloor} \mathbb{P}\{J_t = i\} \cdot 4^i \mathbb{E}_t[\|g_i^t - g_{i-1}^t\|^2] \\ &= 2\mathbb{E}_t[\|\nabla f(x) - g_0^t\|^2] + 2\sum\nolimits_{i=1}^{\lfloor \log_2 M \rfloor} 2^i \mathbb{E}_t[\|g_i^t - g_{i-1}^t\|^2] \\ &\le 2\mathbb{E}_t[\|\nabla f(x) - g_0^t\|^2] \\ &\quad + 4\sum\nolimits_{i=1}^{\lfloor \log_2 M \rfloor} 2^i \left(\mathbb{E}_t[\|\nabla f(x) - g_{i-1}^t\|^2 + \mathbb{E}_t[\|g_i^t - \nabla f(x)\|^2]\right).\end{aligned}$$

To bound $\mathbb{E}_t[\|\nabla f(x) - g_0^t\|^2]$, $\mathbb{E}_t[\|\nabla f(x) - g_{i-1}^t\|^2$, $\mathbb{E}_t[\|g_i^t - \nabla f(x)\|^2]$, we apply Lemma 1 and get

$$\begin{aligned}\mathbb{E}_t[\|\nabla f(x) - g^t\|^2] &\le 88B^{-1}\tau_{\text{mix}}(R - \sqrt{\varepsilon})^2 + 4\sum\nolimits_{i=1}^{\lfloor \log_2 M \rfloor} 2^i \cdot \frac{22\tau_{\text{mix}}}{2^i B}\tau_{\text{mix}}(R - \sqrt{\varepsilon})^2 \\ &\le 176\tau_{\text{mix}}B^{-1}\log_2 M(R - \sqrt{\varepsilon})^2.\end{aligned}$$

Now, using Cauchy-Schwarz inequality and Assumption 2 one can obtain

$$\begin{aligned}\mathbb{E}_t\|g^t\|^2 = \mathbb{E}_t\|g^t + \nabla f(x) - \nabla f(x)\|^2 &\le 2\mathbb{E}_t\|\nabla f(x)\|^2 + 2\mathbb{E}_t\|\nabla f(x) - g^t\|^2 \\ &\le 2(1 + 176B^{-1}\tau_{\text{mix}}\log_2 M)(R - \sqrt{\varepsilon})^2\end{aligned}$$

To show the second part of the statement, we use $\mathbb{E}_t[g^t] = \mathbb{E}_t[g^t_{\lfloor \log_2 M \rfloor}]$ and get

$$\|\nabla f(x_g^t) - \mathbb{E}_t[g^t]\|^2 = \|\nabla f(x^t) - \mathbb{E}_k[g^t_{\lfloor \log_2 M \rfloor}]\|^2 .$$

With Lemma 1 and $2^{\lfloor \log_2 M \rfloor} \ge M/2$, we finish the proof. □

Lemma 3. Let Assumptions 1, 2 and 3 be satisfied. Then for iterates defined in (11), $0 \le \beta_1 < \beta_2 \le 1$ and $(\alpha_t)_{t\in\mathbb{N}}$ defined in (12), it holds that

$$\begin{aligned}\mathbb{E}\left[\sum_{i\in[d]} G_i^t \frac{m_i^t}{\sqrt{v_i^t + \varepsilon}}\right] \ge \frac{1}{2}&\left(\sum_{i\in[d]}\sum_{k=0}^{t-1}\beta_1^k \mathbb{E}\left[\frac{(G_i^{t-k})^2}{\sqrt{\tilde{v}_i^{t,k+1} + \varepsilon}}\right]\right) \\ &- \frac{3R_g}{\sqrt{1-\beta_1}}\left(\sum_{k=0}^{t-1}\left(\frac{\beta_1}{\beta_2}\right)^k \sqrt{k+1}\,\mathbb{E}\left[\left\|U^{t-k}\right\|^2\right]\right) \\ &- \frac{\alpha_t^2 L^2}{4R_g}\sqrt{1-\beta_1}\left(\sum_{l=1}^{t-1}\left\|u^{t-l}\right\|^2 \sum_{k=l}^{t-1}\beta_1^k\sqrt{k}\right) \\ &- \sum_{i\in[d]}\sum_{k=0}^{t-1}\left(\frac{\beta_1}{\beta_2}\right)^k \mathbb{E}\left[\frac{|G_i^{t-k}|}{\sqrt{\mathbb{E}_{t-k}\left[(g_i^{t-k})^2\right] + \varepsilon}}\left|\mathbb{E}_{t-k-1}\left[g_i^{t-k}\right] - G_i^{t-k}\right|\right]\end{aligned} \tag{16}$$

Proof. Let us take an iteration $t \in \mathbb{N}$ for the duration of the proof. We have

$$\begin{aligned}\sum_{i\in[d]} G_i^t \frac{m_i^t}{\sqrt{v_i^t + \varepsilon}} &= \sum_{i\in[d]}\sum_{k=0}^{t-1}\beta_1^k G_i^t \frac{g_i^{t-k}}{\sqrt{v_i^t + \varepsilon}} \\ &= \underbrace{\sum_{i\in[d]}\sum_{k=0}^{t-1}\beta_1^k G_i^{t-k}\frac{g_i^{t-k}}{\sqrt{v_i^t + \varepsilon}}}_{A} + \underbrace{\sum_{i\in[d]}\sum_{k=0}^{t-1}\beta_1^k\left(G_i^t - G_i^{t-k}\right)\frac{g_i^{t-k}}{\sqrt{v_i^t + \varepsilon}}}_{B},\end{aligned} \tag{17}$$

Let us first consider B. Let k be an index such that $0 \leq k \leq t-1$. Using B.2 with

$$\kappa = \frac{\sqrt{1-\beta_1}}{2R_g\sqrt{k+1}},\ x = \left|G_i^t - G_i^{t-k}\right|,\ y = \frac{\left|g_i^{t-k}\right|}{\sqrt{v_i^t+\varepsilon}},$$

where R_g comes from Lemma 2, we have

$$|B| \leq \sum_{i\in[d]}\sum_{k=0}^{t-1}\beta_1^k\left(\frac{\sqrt{1-\beta_1}}{4R_g\sqrt{k+1}}\left(G_i^t - G_i^{t-k}\right)^2 + \frac{R_g\sqrt{k+1}}{\sqrt{1-\beta_1}}\frac{(g_i^{t-k})^2}{\varepsilon+v_i^t}\right). \tag{18}$$

For any $i \in [d]$ it holds that

$$\varepsilon + v_i^t \geq \varepsilon + \beta_2^k v_i^{t-k} \geq \beta_2^k(\varepsilon + v_i^{n-k}).$$

Therefore

$$\frac{(g_i^{t-k})^2}{\varepsilon+v_i^t} \leq \frac{1}{\beta_2^k}(U_i^{t-k})^2. \tag{19}$$

Using the L-smoothness of f given by 1 and the convexity of $\|\cdot\|$, we obtain

$$\begin{aligned}\left\|G^t - G^{t-k}\right\|^2 \leq L^2\left\|x^{t-1} - x^{t-k-1}\right\|^2 &= L^2\left\|\sum_{l=1}^k \alpha_{n-l}u^{t-l}\right\|^2 \\ &\leq \alpha_t^2 L^2 k\sum_{l=1}^k\left\|u_{t-l}\right\|^2,\end{aligned} \tag{20}$$

Injecting (19) and (20) into (18), we obtain

$$\begin{aligned}|B| \leq &\left(\sum_{k=0}^{t-1}\frac{\alpha_t^2L^2}{4R_g}\sqrt{1-\beta_1}\beta_1^k\sqrt{k}\sum_{l=1}^k\left\|u^{t-l}\right\|^2\right) \\ &+\left(\sum_{k=0}^{t-1}\frac{R_g}{\sqrt{1-\beta_1}}\left(\frac{\beta_1}{\beta_2}\right)^k\sqrt{k+1}\left\|U^{t-k}\right\|^2\right) &(21)\\ &= \sqrt{1-\beta_1}\frac{\alpha_t^2L^2}{4R_g}\left(\sum_{l=1}^{t-1}\left\|u^{t-l}\right\|^2\sum_{k=l}^{t-1}\beta_1^k\sqrt{k}\right) &(22)\\ &+\frac{R_g}{\sqrt{1-\beta_1}}\left(\sum_{k=0}^{t-1}\left(\frac{\beta_1}{\beta_2}\right)^k\sqrt{k+1}\left\|U^{t-k}\right\|^2\right). &(23)\end{aligned}$$

Now going back to the A term from (17). We will further drop indices for some part of the proof, noting

$$G := G_i^{t-k}, g := g_i^{t-k}, \tilde{v} := \tilde{v}_i^{t,k+1} \text{ and } v := v_i^t.$$

Finally, let us make an auxiliary notation

$$\delta^2 := \sum_{j=t-k}^{t}\beta_2^{t-j}(g_i^j)^2 \qquad \text{and} \qquad r^2 := \mathbb{E}_{t-k-1}\left[\delta^2\right]. \tag{24}$$

In particular, we have $\tilde{v} - v = r^2 - \delta^2$. With our new notations, we obtain that:

$$\begin{aligned}
\mathbb{E}\left[G\frac{g}{\sqrt{v+\varepsilon}}\right] &= \mathbb{E}\left[G\frac{g}{\sqrt{\tilde{v}+\varepsilon}} + Gg\left(\frac{1}{\sqrt{v+\varepsilon}} - \frac{1}{\sqrt{\tilde{v}+\varepsilon}}\right)\right] \\
&= \mathbb{E}\left[\mathbb{E}_{t-k-1}\left[G\frac{g}{\sqrt{\tilde{v}+\varepsilon}}\right] + Gg\frac{r^2-\delta^2}{\sqrt{v+\varepsilon}\sqrt{\tilde{v}+\varepsilon}(\sqrt{v+\varepsilon}+\sqrt{\tilde{v}+\varepsilon})}\right] \\
&= \mathbb{E}\left[\frac{G^2}{\sqrt{\tilde{v}+\varepsilon}}\right] + \underbrace{\mathbb{E}\left[\frac{G(\mathbb{E}_{t-k-1}[g]-G)}{\sqrt{\tilde{v}+\varepsilon}}\right]}_{C} \\
&\quad + \mathbb{E}\left[\underbrace{Gg\frac{r^2-\delta^2}{\sqrt{v+\varepsilon}\sqrt{\tilde{v}+\varepsilon}(\sqrt{v+\varepsilon}+\sqrt{\tilde{v}+\varepsilon})}}_{D}\right].
\end{aligned} \tag{25}$$

Consider C. Since $\tilde{v} \geq \beta_2^k \mathbb{E}_{t-k}\left[g^2\right]$ we obtain:

$$\mathbb{E}\left[\left(\frac{G(\mathbb{E}_{t-k-1}[g]-G)}{\sqrt{\tilde{v}+\varepsilon}}\right)^2\right] \leq \frac{1}{\beta_2^{2k}}\mathbb{E}\left[\frac{G^2}{\varepsilon+\mathbb{E}_{t-k}[g^2]}(\mathbb{E}_{t-k-1}[g]-G)^2\right].$$

Consider D.

$$|C| \leq \underbrace{|Gg|\frac{r^2}{\sqrt{v+\varepsilon}(\varepsilon+\tilde{v})}}_{\textcircled{1}} + \underbrace{|Gg|\frac{\delta^2}{(\varepsilon+v)\sqrt{\tilde{v}+\varepsilon}}}_{\textcircled{2}},$$

due to the fact that $(a_1 - a_2)/(a_3 + a_4) \leq a_1/a_3 + a_2/a_4$ for all $a_{1,2,3,4} \geq 0$. Consider ①. Applying B.2 with

$$\kappa = \frac{\sqrt{1-\beta_1}\sqrt{\tilde{v}+\varepsilon}}{2},\ x = \frac{|G|}{\sqrt{\tilde{v}+\varepsilon}},\ y = \frac{|g|\,r^2}{\sqrt{\tilde{v}+\varepsilon}\sqrt{v+\varepsilon}},$$

we obtain

$$\textcircled{1} \leq \frac{G^2}{4\sqrt{\tilde{v}+\varepsilon}} + \frac{1}{\sqrt{1-\beta_1}}\frac{g^2 r^4}{(\varepsilon+\tilde{v})^{3/2}(\varepsilon+v)}.$$

Given that $\varepsilon + \tilde{v} \geq r^2$ and taking the conditional expectation, we can simplify as

$$\mathbb{E}_{t-k-1}\left[\textcircled{1}\right] \leq \frac{G^2}{4\sqrt{\tilde{v}+\varepsilon}} + \frac{1}{\sqrt{1-\beta_1}}\frac{r^2}{\sqrt{\tilde{v}+\varepsilon}}\mathbb{E}_{n-k-1}\left[\frac{g^2}{\varepsilon+v}\right]. \tag{26}$$

Now consider ②. Using B.2 with

$$\kappa = \frac{\sqrt{1-\beta_1}\sqrt{\tilde{v}+\varepsilon}}{2r^2},\ x = \frac{|G\delta|}{\sqrt{\tilde{v}+\varepsilon}},\ y = \frac{|\delta g|}{\varepsilon+v},$$

we obtain

$$\textcircled{2} \leq \frac{G^2}{4\sqrt{\tilde{v}+\varepsilon}}\frac{\delta^2}{r^2} + \frac{1}{\sqrt{1-\beta_1}}\frac{r^2}{\sqrt{\tilde{v}+\varepsilon}}\frac{g^2\delta^2}{(\varepsilon+v)^2}. \tag{27}$$

Given that $\varepsilon + v \geq \delta^2$, and $\mathbb{E}_{n-k-1}\left[\delta^2/r^2\right] = 1$, we obtain after taking the conditional expectation,

$$\mathbb{E}_{t-k-1}\left[\textcircled{2}\right] \leq \frac{G^2}{4\sqrt{\tilde{v}+\varepsilon}} + \frac{1}{\sqrt{1-\beta_1}}\frac{r^2}{\sqrt{\tilde{v}+\varepsilon}}\mathbb{E}_{n-k-1}\left[\frac{g^2}{\varepsilon+v}\right]. \tag{28}$$

Summing (26) and (28), we get

$$\mathbb{E}_{t-k-1}\left[|D|\right] \leq \frac{G^2}{2\sqrt{\tilde{v}+\varepsilon}} + \frac{2}{\sqrt{1-\beta_1}}\frac{r^2}{\sqrt{\tilde{v}+\varepsilon}}\mathbb{E}_{n-k-1}\left[\frac{g^2}{\varepsilon+v}\right]. \tag{29}$$

Given that $r \leq \sqrt{\tilde{v} + \varepsilon}$ by definition of $\tilde{v}$, and that using Lemma 2, $r \leq \sqrt{k+1} R_g$, we have, reintroducing the indices we had dropped

$$\mathbb{E}_{t-k-1}\left[|D|\right] \leq \frac{(G_i^{t-k})^2}{2\sqrt{\tilde{v}_i^{t,k+1} + \varepsilon}} + \frac{2R_g}{\sqrt{1-\beta_1}} \sqrt{k+1} \mathbb{E}_{t-k-1}\left[\frac{(g_i^{t-k})^2}{\varepsilon + v_i^t}\right]. \tag{30}$$

Taking the complete expectation and using that by definition $\varepsilon + v_i^t \geq \varepsilon + \beta_2^k v_i^{t-k} \geq \beta_2^k(\varepsilon + v_i^{t-k})$ we get

$$\mathbb{E}\left[|D|\right] \leq \frac{1}{2}\mathbb{E}\left[\frac{(G_i^{t-k})^2}{\sqrt{\tilde{v}_i^{t,k+1} + \varepsilon}}\right] + \frac{2R_g}{\sqrt{1-\beta_1}\beta_2^k} \sqrt{k+1} \mathbb{E}\left[\frac{(g_i^{t-k})^2}{\varepsilon + v_i^{t-k}}\right]. \tag{31}$$

Injecting (31) into (25) and using B.2 gives us

$$\begin{aligned}
\mathbb{E}\left[A\right] \geq \sum_{i\in[d]} \sum_{k=0}^{t-1} \beta_1^k \Bigg\{ & \mathbb{E}\left[\frac{(G_i^{t-k})^2}{\sqrt{\tilde{v}_i^{t,k+1} + \varepsilon}}\right] \\
& - \frac{1}{\beta_2^k}\mathbb{E}\left[\frac{|G_i^{t-k}|}{\sqrt{\mathbb{E}_{t-k}\left[(g_i^{t-k})^2\right] + \varepsilon}} \left|\mathbb{E}_{t-k-1}\left[g_i^{t-k}\right] - G_i^{t-k}\right|\right] \\
& - \left(\frac{1}{2}\mathbb{E}\left[\frac{(G_i^{t-k})^2}{\sqrt{\tilde{v}_i^{t,k+1} + \varepsilon}}\right] + \frac{2R_g}{\sqrt{1-\beta_1}\beta_2^k}\sqrt{k+1}\mathbb{E}\left[\frac{(g_i^{t-k})^2}{\varepsilon + v_i^{t-k}}\right]\right)\Bigg\} \\
= \frac{1}{2}\left(\sum_{i\in[d]} \sum_{k=0}^{t-1} \beta_1^k \mathbb{E}\left[\frac{(G_i^{t-k})^2}{\sqrt{\tilde{v}_i^{t,k+1} + \varepsilon}}\right]\right) & - \frac{2R_g}{\sqrt{1-\beta_1}}\left(\sum_{k=0}^{t-1}\left(\frac{\beta_1}{\beta_2}\right)^k \sqrt{k+1}\mathbb{E}\left[\left\|U^{t-k}\right\|^2\right]\right) \\
- \sum_{i\in[d]} \sum_{k=0}^{t-1} \left(\frac{\beta_1}{\beta_2}\right)^k \mathbb{E}\left[\frac{|G_i^{t-k}|}{\sqrt{\mathbb{E}_{t-k}\left[(g_i^{t-k})^2\right] + \varepsilon}} \left|\mathbb{E}_{t-k-1}\left[g_i^{t-k}\right] - G_i^{t-k}\right|\right]&.
\end{aligned} \tag{32}$$

Injecting (32) and (23) into (17) finishes the proof. □

C.3 Proof of the main theorem

Theorem 2 (Theorem 1)**.** Let Assumptions 1, 2, 3 be satisfied. Given the iterates defined by system (11), $0 \leq \beta_1 < \beta_2 < 1$, $\alpha_t = \alpha(1-\beta_1)\sqrt{\frac{1-\beta_2^t}{1-\beta_2}}$ and $N(T)$ defined by (3), for any T such that $T > \max\{\frac{\beta_1}{1-\beta_1} \ ; \ \tau_{\text{mix}}\}$ it holds that

$$\begin{aligned}
\mathbb{E}\left[\left\|\nabla f(x^{N(T)})\right\|^2\right] = \mathcal{O}\Bigg(& R_g \frac{f(x^0) - f^*}{\alpha \tilde{T}} + \frac{\Delta}{\tilde{T}}\left[\ln\left(1 + \frac{R_g^2}{\varepsilon(1-\beta_2)}\right) - T\log(\beta_2)\right] \\
& + \frac{dR_g\mu_g}{\tilde{T}}\left[T - \tau_{\text{mix}} + \left(\frac{\beta_1}{\beta_2}\right)^{-\tau_{\text{mix}}/2}\Delta_{\tau_{\text{mix}}}\right]\frac{1-\beta_1}{(1-\beta_1/\beta_2)\sqrt{1-\beta_2}}\Bigg),
\end{aligned}$$

where $R_g^2 := (1 + B^{-1}\tau_{\text{mix}} \log M)R^2$, $\mu_g^2 := B^{-1}\tau_{\text{mix}}M^{-1}R^2$ come from Lemma 2, $\tilde{T} := T - \beta_1/(1-\beta_1)$ and

$$\begin{aligned}
\Delta &:= \frac{\alpha d R_g L(1-\beta_1)}{(1-\beta_1/\beta_2)(1-\beta_2)} + \frac{2\alpha^2 d L^2 \beta_1}{(1-\beta_1/\beta_2)(1-\beta_2)^{3/2}} + \frac{12 d R_g^2\sqrt{1-\beta_1}}{(1-\beta_1/\beta_2)^{3/2}\sqrt{1-\beta_2}}, \\
\Delta_{\tau_{\text{mix}}} &:= \sum_{t=1}^{\tau_{\text{mix}}} \frac{\|\nabla f(x^t)\|}{\sqrt{\mathbb{E}_t\left[\|g^t\|^2\right] + \varepsilon}}.
\end{aligned}$$

Proof. Let us a take an iteration $t \in \mathbb{N}$. Using the smoothness of f defined in Assumption 1, we have

$$f(x^t) \leq f(x^{t-1}) - \alpha_t \left\langle G^t, u^t \right\rangle + \frac{\alpha_t^2 L}{2} \left\| u^t \right\|^2 .$$

Taking the full expectation and using Lemma 3,

$$\begin{aligned}
\mathbb{E}\left[f(x^t)\right] \leq \mathbb{E}\left[f(x^{t-1})\right] &- \frac{\alpha_t}{2}\left(\sum_{i\in[d]}\sum_{k=0}^{t-1}\beta_1^k \mathbb{E}\left[\frac{(G_i^{t-k})^2}{\sqrt{\varepsilon + \tilde{v}_i^{t+1}}}\right]\right) + \frac{\alpha_t^2 L}{2}\mathbb{E}\left[\left\|u^t\right\|_2^2\right] \\
&+ \frac{\alpha_t^3 L^2}{4R_g}\sqrt{1-\beta_1}\left(\sum_{l=1}^{t-1}\left\|u^{t-l}\right\|_2^2 \sum_{k=l}^{t-1}\beta_1^k\sqrt{k}\right) \\
&+ \frac{3\alpha_t R_g}{\sqrt{1-\beta_1}}\left(\sum_{k=0}^{t-1}\left(\frac{\beta_1}{\beta_2}\right)^k \sqrt{k+1}\left\|U^{t-k}\right\|^2\right) \\
&+ \alpha_t \sum_{i\in[d]}\sum_{k=0}^{t-1}\left(\frac{\beta_1}{\beta_2}\right)^k \mathbb{E}\left[\frac{|G_i^{t-k}|}{\sqrt{\mathbb{E}_{t-k}\left[(g_i^{t-k})^2\right]+\varepsilon}}\left|\mathbb{E}_{t-k-1}\left[g_i^{t-k}\right] - G_i^{t-k}\right|\right] \qquad (33)
\end{aligned}$$

We now consider term of the form $\sum_{i\in[d]} \beta_1^k \mathbb{E}\left[\frac{(G_i^{t-k})^2}{\sqrt{\varepsilon+\tilde{v}_i^{t+1}}}\right]$. Introducing notation $\tilde{V}_{t,k+1} := \sum_{i\in[d]} \tilde{v}_i^{t,k+1}$ we obtain that

$$\sum_{i\in[d]} \beta_1^k \mathbb{E}\left[\frac{(G_i^{t-k})^2}{\sqrt{\varepsilon + \tilde{v}_i^{t+1}}}\right] \geq \beta_1^k \mathbb{E}\left[\frac{(G_i^{t-k})^2}{\sqrt{\varepsilon + \tilde{V}_{t,k+1}}}\right].$$

Taking $X := \left(\frac{\left\|G^{n-k}\right\|^2}{\sqrt{\varepsilon+\tilde{V}_{t,k+1}}}\right)^{\frac{2}{3}}$, $Y := \left(\sqrt{\varepsilon + \tilde{V}_{t,k+1}}\right)^{\frac{2}{3}}$, we can apply Hölder inquality B.1 as

$$\mathbb{E}\left[|X|^{\frac{3}{2}}\right] \geq \left(\frac{\mathbb{E}\left[|XY|\right]}{\mathbb{E}\left[|Y|^3\right]^{\frac{1}{3}}}\right)^{\frac{3}{2}}, \qquad (34)$$

which gives us

$$\mathbb{E}\left[\frac{\left\|G^{t-k}\right\|^2}{\sqrt{\epsilon + \tilde{V}_{n,k+1}}}\right] \geq \frac{\mathbb{E}\left[\left\|G^{t-k}\right\|^{\frac{4}{3}}\right]^{\frac{3}{2}}}{\sqrt{\mathbb{E}\left[\epsilon + \tilde{V}_{n,k+1}\right]}} \geq \frac{\mathbb{E}\left[\left\|G^{t-k}\right\|^{\frac{4}{3}}\right]^{\frac{3}{2}}}{\Omega_t R_g}, \qquad (35)$$

with $\Omega_t := \sqrt{\sum_{j=0}^{t-1}\beta_2^j}$, and using the fact that $\mathbb{E}\left[\varepsilon + \sum_{i\in[d]} \tilde{v}_i^{t,k+1}\right] \leq R_g^2 \Omega_t^2$.

Now sum (33) over all iterations $t \in [T]$ and using (35) we can obtain that

$$\begin{aligned}
&\underbrace{\frac{1}{2R_g}\sum_{t=1}^{T}\frac{\alpha_t}{\Omega_t}\sum_{k=0}^{n-1}\beta_1^k\mathbb{E}\left[\left\|G^{t-k}\right\|^{\frac{4}{3}}\right]^{\frac{3}{2}}}_{A} \leq f(x^0) - f^* + \underbrace{\frac{\alpha_T^2 L}{2}\sum_{t=1}^{T}\mathbb{E}\left[\left\|u^t\right\|^2\right]}_{B} \\
&+ \underbrace{\frac{\alpha_T^3 L^2}{4R_g}\sqrt{1-\beta_1}\sum_{t=1}^{T}\sum_{l=1}^{n-1}\mathbb{E}\left[\left\|u^{t-l}\right\|^2\right]\sum_{k=l}^{n-1}\beta_1^k\sqrt{k}}_{C} \\
&+ \underbrace{\frac{3\alpha_T R_g}{\sqrt{1-\beta_1}}\sum_{t=1}^{T}\sum_{k=0}^{t-1}\left(\frac{\beta_1}{\beta_2}\right)^k\sqrt{k+1}\mathbb{E}\left[\left\|U^{t-k}\right\|^2\right]}_{D} \\
&+ \underbrace{\sum_{t=1}^{T}\alpha_t\sum_{i\in[d]}\sum_{k=0}^{t-1}\left(\frac{\beta_1}{\beta_2}\right)^k\mathbb{E}\left[\frac{|G_i^{t-k}|}{\sqrt{\mathbb{E}_{t-k}\left[(g_i^{t-k})^2\right]+\varepsilon}}\left|\mathbb{E}_{t-k-1}\left[g_i^{t-k}\right]-G_i^{t-k}\right|\right]}_{E}. \qquad (36)
\end{aligned}$$

Since our estimates A, B, C, D coincide with corresponding estimates from Défossez et al. (2020) we utilize the following result

$$A \geq \frac{\alpha\tilde{T}}{2R_g}\mathbb{E}\left[\left\|\nabla f(x^{N(T)})\right\|^{4/3}\right]^{3/2}, \qquad (37)$$

$$B \leq \frac{d\alpha_T^2 L}{2(1-\beta_1)(1-\beta_1/\beta_2)}\left(\ln\left(1+\frac{R_g^2}{\varepsilon(1-\beta_2)}\right) - T\log(\beta_2)\right), \qquad (38)$$

$$C \leq \frac{d\alpha_T^3 L^2\beta_1}{R_g(1-\beta_1)^3(1-\beta_1/\beta_2)}\left(\ln\left(1+\frac{R_g^2}{\varepsilon(1-\beta_2)}\right) - T\log(\beta_2)\right), \qquad (39)$$

$$D \leq \frac{6d\alpha_T R_g}{\sqrt{1-\beta_1}(1-\beta_1/\beta_2)^{3/2}}\left(\ln\left(1+\frac{R_g}{\varepsilon(1-\beta_2)}\right) - T\ln(\beta_2)\right), \qquad (40)$$

where $N(T)$ sampled from (3), α_T, α come from (12) and

$$\tilde{T} := T - \frac{\beta_1}{1-\beta_1}.$$

For (37) see equation (A.46), for (38) see (A.38), for (39) see (A.40), for (40) see (A.42) from Défossez et al. (2020).

Now consider E. By definition of f we obtain that

$$\frac{(G_i^{t-k})^2}{\mathbb{E}_{t-k}\left[(g_i^{t-k})^2\right]+\varepsilon} = \frac{\left(\mathbb{E}_{Z\sim\mu}\left[\nabla f(x^{t-k-1}, Z)_i\right]\right)^2}{\mathbb{E}_{t-k}\left[(g_i^{t-k})^2\right]+\varepsilon},$$

where μ is a stationary distribution of the Markov chain $\{Z_t\}_{t=0}^{\infty}$. Now assuming that $g^{t-k}(Z)$ is the same as g^{t-k} in line 5 of Algorithm 1, but $Z \sim \mu$ is used instead of Z_{T^k}. Now we again drop indexes and using Assumption 3 we obtain

$$\begin{aligned}
\frac{\left(\mathbb{E}_{Z\sim\mu}\left[\nabla f(x,Z)\right]\right)^2}{\mathbb{E}_{t-k}\left[g^2\right]+\varepsilon} = \frac{\left(\mathbb{E}_{Z\sim\mu}\left[g(Z)\right]\right)^2}{\mathbb{E}_{t-k}\left[g^2\right]+\varepsilon} &\le \frac{\mathbb{E}_{Z\sim\mu}\left[g(Z)^2\right]}{\mathbb{E}_{t-k}\left[g^2\right]+\varepsilon} \\
&\le \frac{\sum_{z\in\mathcal{Z}} g(z)^2\mu_z}{\sum_{z\in\mathcal{Z}} g(z)^2\mathbb{P}\left\{Z_{t-k}=z\right\}+\varepsilon} \\
&= \frac{\sum_{z\in\mathcal{Z}} g(z)^2\mathbb{P}\left\{Z_{t-k}=z\right\}+\sum_{z\in\mathcal{Z}} g(z)^2(\mu_z-\mathbb{P}\left\{Z_{t-k}=z\right\})}{\sum_{z\in\mathcal{Z}} g(z)^2\mathbb{P}\left\{Z_{t-k}=z\right\}+\varepsilon} \\
&\le \frac{\mathbb{E}_{t-k}\left[g(Z)^2\right]+\sum_{z\in\mathcal{Z}} g(z)^2|\mu_z-\mathbb{P}\left\{Z_{t-k}=z\right\}|}{\mathbb{E}_{t-k}\left[g(Z)^2\right]+\varepsilon} \\
&\le \frac{\mathbb{E}_{t-k}\left[g(Z)^2\right]+R_g^2/\mu_{\min}\cdot(1/2)^{(t-k)/\tau_{\text{mix}}}}{\mathbb{E}_{t-k}\left[g(Z)^2\right]+\varepsilon},
\end{aligned}$$

where $\mu_{\min} := \min_{z\in\mathcal{Z}}\{\mu_z\}$. Consider $t-k \ge \log(R_g^2/(\mu_{\min}\varepsilon))\tau_{\text{mix}} \gtrsim \tau_{\text{mix}}$, then we obtain result of the form

$$\frac{\left(\mathbb{E}_{Z\sim\mu}\left[\nabla f(x,Z)\right]\right)^2}{\mathbb{E}_{t-k}\left[g^2\right]+\varepsilon} \le 1.$$

For $t-k \lesssim \tau_{\text{mix}}$, then we just define the notation of the form

$$\Delta_{\tau_{\text{mix}}} := \sum_{t=1}^{\tau_{\text{mix}}} \frac{\|\nabla f(x^t)\|}{\sqrt{\mathbb{E}_t\left[\|g^t\|^2\right]+\varepsilon}}.$$

Then we obtain that

$$\begin{aligned}
E \le \alpha_T\sqrt{d}\, &\underbrace{\sum_{t=\tau_{\text{mix}}}^{T}\sum_{k=0}^{t-\tau_{\text{mix}}}\left(\frac{\beta_1}{\beta_2}\right)^k\mu_g}_{\textcircled{1}} \\
&+ \alpha_T d \underbrace{\sum_{t=0}^{T}\sum_{k=\max\{0;t-\tau_{\text{mix}}\}}^{t-1}\left(\frac{\beta_1}{\beta_2}\right)^k\frac{\left\|\nabla f(x^{t-k})\right\|}{\sqrt{\mathbb{E}_t\left[\|g^{t-k}\|^2\right]+\varepsilon}}\mu_g}_{\textcircled{2}},
\end{aligned}$$

where μ_g comes from Lemma 2. Consider $\textcircled{1}$:

$$\textcircled{1} \le \frac{\mu_g(T-\tau_{\text{mix}})}{1-\beta_1/\beta_2}.$$

Consider $\textcircled{2}$. Changing indexes as $l=t-k$ and $q=t+k$ we obtain

$$\begin{aligned}
\textcircled{2} &= \sum_{q=0}^{2T}\sum_{l=0}^{\tau_{\text{mix}}}\left(\frac{\beta_1}{\beta_2}\right)^{(q-l)/2}\frac{\left\|\nabla f(x^l)\right\|}{\sqrt{\mathbb{E}_t\left[\|g^l\|^2\right]+\varepsilon}}\mu_g \\
&\le \mu_g\left(\frac{\beta_1}{\beta_2}\right)^{-\tau_{\text{mix}}/2}\left(\sum_{q=0}^{2T}\left(\frac{\beta_1}{\beta_2}\right)^{q/2}\right)\cdot\left(\sum_{l=0}^{\tau_{\text{mix}}}\frac{\left\|\nabla f(x^l)\right\|}{\sqrt{\mathbb{E}_t\left[\|g^l\|^2\right]+\varepsilon}}\right) \\
&\le \mu_g\left(\frac{\beta_1}{\beta_2}\right)^{-\tau_{\text{mix}}/2}\frac{\Delta_{\tau_{\text{mix}}}}{1-\beta_1/\beta_2}.
\end{aligned}$$

Therefore we can estimate E:

$$E \le d\alpha_T\left(T-\tau_{\text{mix}}+\left(\frac{\beta_1}{\beta_2}\right)^{-\tau_{\text{mix}}/2}\Delta_{\tau_{\text{mix}}}\right)\frac{\mu_g}{1-\beta_1/\beta_2}.$$

Combining all estimates on A, B, C, D, E we obtain:

$$\mathbb{E}\left[\left\|\nabla f(x^{N(T)})\right\|^{4/3}\right]^{3/2} = \mathcal{O}\Bigg(R_g\frac{f(x^0)-f^*}{\alpha\tilde{T}} + \frac{\Delta}{\tilde{T}}\left[\ln\left(1+\frac{R_g^2}{\varepsilon(1-\beta_2)}\right) - T\log(\beta_2)\right]$$

$$+\frac{dR_g\mu_g}{\tilde{T}}\left[T-\tau_{\text{mix}}+\left(\frac{\beta_1}{\beta_2}\right)^{-\tau_{\text{mix}}/2}\Delta_{\tau_{\text{mix}}}\right]\frac{1-\beta_1}{(1-\beta_1/\beta_2)\sqrt{1-\beta_2}}\Bigg),$$

where

$$\Delta := \frac{\alpha dR_gL(1-\beta_1)}{(1-\beta_1/\beta_2)(1-\beta_2)} + \frac{2\alpha^2dL^2\beta_1}{(1-\beta_1/\beta_2)(1-\beta_2)^{3/2}} + \frac{12dR_g^2\sqrt{1-\beta_1}}{(1-\beta_1/\beta_2)^{3/2}\sqrt{1-\beta_2}}.$$

This finishes the proof. □