Iranian Journal of Numerical Analysis and Optimization
Vol. 16, No. 2, 2026, pp 727–756
https://doi.org/10.22067/ijnao.2026.96244.1767
https://ijnao.um.ac.ir/

Research Article

Tailed-Average SGD: A Polyak–Ruppert modification for optimal-rate constrained optimization

A. Fillali*, S. Addoune and R. Zeghdane

Abstract

Stochastic gradient descent (SGD) algorithms are a cornerstone of large-scale optimization, yet their application to problems with explicit constraints remains a significant challenge. While methods based on relaxed barrier functions offer a promising alternative that avoids computationally expensive projections, they often suffer from suboptimal convergence. Due to

*Corresponding author

Received 30 October 2025; revised 19 February 2026; accepted 21 February 2026

Abdelouakil Fillali
Laboratoire d'Analyse Mathématiques et ses Applications (LAMA), Department of Mathematics, University Mohamed El Bachir El Ibrahimi, El Anasser, Bordj Bou Arreridj, 34030, Algeria. e-mail: abdelouakil.fillali@univ-bba.dz

Smail Addoune
Laboratoire d'Analyse Mathématiques et ses Applications (LAMA), Department of Mathematics, University Mohamed El Bachir El Ibrahimi, El Anasser, Bordj Bou Arreridj, 34030, Algeria. e-mail: smail.addoune@univ-bba.dz

Rebiha Zeghdane
Laboratoire d'Analyse Mathématiques et ses Applications (LAMA), Department of Mathematics, University Mohamed El Bachir El Ibrahimi, El Anasser, Bordj Bou Arreridj, 34030, Algeria. e-mail: rebiha.zeghdane@univ-bba.dz

How to cite this article
Fillali, A., Addoune, S. and Zeghdane, R., Tailed-Average SGD: A Polyak–Ruppert modification for optimal-rate constrained optimization. *Iran. J. Numer. Anal. Optim.*, 2026; 16(2): 727-756. https://doi.org/10.22067/ijnao.2026.96244.1767

the inherent stochastic noise, their convergence stagnates at a certain level of accuracy, failing to achieve the optimal rate.

This paper proposes Tailed-Average SGD (TA-SGD), a novel modification designed to recover this optimal rate. Our algorithm integrates the classical Polyak–Ruppert averaging technique, but critically, it applies this averaging *only after* a predefined initial "burn-in" period (K_{burn}).

This "tailed" approach strategically discards the initial, high-error iterates from the transient phase, which we demonstrate can "poison" a naive averaging scheme applied from $k = 0$. We provide a rigorous theoretical analysis proving that TA-SGD achieves the best of both worlds: It preserves the almost sure convergence guarantees of the baseline method, while simultaneously succeeding in noise cancellation to achieve the optimal mean squared error (MSE) rate of $O(1/K)$.

Our numerical experiments on a large-scale quadratic program validate these theoretical claims, demonstrating that TA-SGD successfully overcomes this stagnation. It significantly outperforms both the baseline algorithm (which stagnates) and the naive full-averaging scheme (which converges prohibitively slowly).

AMS subject classifications (2020): Primary 90C15, 90C25; Secondary 49M30, 65K05, 68T05.

Keywords: Constrained optimization, Polyak–Ruppert averaging, Adaptive barrier function, Stochastic gradient descent, Optimal rate.

1 Introduction

Stochastic gradient descent (SGD) and its variants have become cornerstone algorithms in modern large-scale optimization, driving progress in fields ranging from machine learning to networked systems [1, 13, 24]. These methods are fundamentally rooted in the pioneering work on stochastic approximation by Robbins and Monro [16].

While SGD is exceptionally efficient for unconstrained problems [7], its application to constrained optimization remains an active area of research [19]. Many classical approaches require full knowledge of the constraint set at each iteration, often necessitating computationally expensive projection steps that are prohibitive when the number of constraints is very large [21].

To overcome this challenge, recent methods have been developed that rely on sampling only a subset of constraints. These include methods based on penalty functions [11, 22] or primal-dual approaches [23]. A particularly promising direction, however, is the Adaptive Relaxed Barrier SGD algorithm proposed by Dimitrieski, Cao, and Ebenbauer [4]. This approach avoids

expensive projections by using a carefully adapted relaxed barrier function [6, 5], and it allows for sampling both the objective function and the constraints.

While this baseline algorithm is guaranteed to converge almost surely [4], our investigation reveals a critical limitation: Its convergence stagnates at a suboptimal level. Due to the inherent stochastic noise, the algorithm fails to achieve the optimal convergence rate.

To address these limitations, this paper proposes tailed-average SGD (TA-SGD).

Contributions and Novelty.

It is important to distinguish our contribution from the baseline work of Dimitrieski, Cao, and Ebenbauer [4]. While they established the almost sure convergence of the adaptive relaxed barrier SGD, their analysis and experiments show that the algorithm suffers from variance-induced stagnation, preventing it from reaching the optimal $O(1/K)$ rate. Our work differs in two key aspects:

1. **Algorithmic Modification:** We introduce a "Tailed-Averaging" mechanism with a strictly defined burn-in phase (K_{burn}). Unlike the standard Polyak–Ruppert averaging [15] applied from the start (which we show fails due to the "poisoning effect" of transient iterates), our approach specifically targets the post-transient phase.

2. **Theoretical Guarantee:** We provide a rigorous proof (Section 3) demonstrating that this modification recovers the optimal convergence rate established by Polyak and Juditsky [15] and reduces the asymptotic variance, a result not achieved in the baseline work [4].

Our numerical experiments on a large-scale quadratic program validate these theoretical claims, demonstrating that TA-SGD successfully overcomes the stagnation and significantly outperforms both the baseline algorithm and the naive full-averaging scheme.

2 Proposed algorithm: Tailed-average SGD

In this section, we introduce our proposed algorithm, named **TA-SGD (Tailed-average SGD)**. This algorithm is designed to efficiently solve large-scale constrained optimization problems by combining the adaptive relaxed barrier method of Dimitrieski, Cao, and Ebenbauer [4], with the optimal-rate acceleration technique of Polyak–Ruppert averaging [15].

The core idea is to first use the Adaptive Relaxed Barrier SGD [4] to generate a "primary iterate" (x_k) that handles constraints stochastically. Then, to achieve an optimal convergence

rate and overcome the high-variance "stagnation" issue inherent in standard SGD, we apply the Polyak–Ruppert averaging technique, but only after an initial "burn-in" period.

2.1 Problem formulation and assumptions

The objective is to solve the following constrained optimization problem:

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x) \quad \text{subject to} \quad g_j(x) \le 0, \quad \text{for all } j \in \{1,\dots,m\}, \tag{1}$$

where we assume that the objective functions f_i and constraint functions g_j are continuously differentiable ($\mathcal{C}^1$) functions. We adopt the following standard assumptions, with mathematical formulations that are common in this field [3, 2, 18]:

Assumption 1 (Objective function properties)**.** The objective function f and its components f_i are assumed to possess the following properties for constants $L \ge \theta > 0$ and for any $x, y \in \mathbb{R}^d$:

1. ***L*-Smoothness:** Each component $f_i(x)$ is L-smooth, meaning its gradient is Lipschitz continuous with constant L:

$$f_i(y) \le f_i(x) + \nabla f_i(x)^T(y-x) + \frac{L}{2}||y-x||^2.$$

2. **Convexity:** Each component $f_i(x)$ is convex:

$$f_i(y) \ge f_i(x) + \nabla f_i(x)^T(y-x).$$

3. **Strong Convexity:** We assume that at least one component function $f_i(x)$ is θ-strongly convex. Consequently, the average function $f(x)$ is strongly convex with parameter θ:

$$f(y) \ge f(x) + \nabla f(x)^T(y-x) + \frac{\theta}{2}\|y-x\|^2.$$

These assumptions, although restrictive, are not uncommon in convergence proofs for gradient-based optimization algorithms; see [2, 4, 13].

Assumption 2 (Constraint functions)**.** The constraint functions $g_j(x)$ are affine, defined as follows:

$$g_j(x) := a_j^T x + b_j,$$

where $a_j \in \mathbb{R}^d$ and $b_j \in \mathbb{R}$. The feasible set $\mathbb{X} := \{x \in \mathbb{R}^d : g_j(x) \leq 0, \text{ for all } j\}$ is assumed to have a nonempty interior, which is a fundamental condition for applying barrier-based methods [2, 6]

Remark 1 (Extension to nonaffine constraints)**.** It is important to note that while our formal convergence analysis (Theorem 1) relies on the affine nature of $g_j(x)$ to guarantee the quadratic upper bound of the barrier function, the proposed TA-SGD framework is theoretically extensible to general convex constraints. As discussed in the foundational framework [4], Assumption 2 can be relaxed to include any continuously differentiable ($\mathcal{C}^1$) convex functions, provided that the iterates $\{x_k\}$ are ensured to remain bounded (e.g., via projection onto a large compact set). The "tailed-averaging" mechanism we propose is particularly beneficial in such nonaffine settings, as it effectively dampens the higher variance introduced by the curvature of nonlinear constraints.

Assumption 3 (Stochastic sampling)**.** At each iteration k, an objective index $i_k \sim \mathcal{U}\{1, \ldots, n\}$ and a constraint index $j_k \sim \mathcal{U}\{1, \ldots, m\}$ are sampled independently and uniformly at random. This uniform distribution is chosen to ensure that the stochastic gradients are unbiased estimators of the true gradients, thereby satisfying the standard assumptions for SGD algorithms [16]. This is demonstrated by the following equations:

1. **For the objective function:** The expected value of the sampled stochastic gradient is equal to the true gradient of the full function [4, 16]:

$$\mathbb{E}[\nabla f_{i_k}(x)] = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x) = \nabla f(x).$$

2. **For the barrier function:** The expected value of the sampled stochastic gradient of the barrier function equals the average effect of the gradients over all constraints [4]:

$$\mathbb{E}[\nabla B(g_{j_k}(x), \delta)] = \frac{1}{m}\sum_{j=1}^{m} \nabla B(g_j(x), \delta).$$

Before describing the algorithm, we formally define the **"Relaxed Problem"** associated with the barrier parameter δ. It is the unconstrained minimization of the objective function augmented by the relaxed barrier:

$$\min_{x \in \mathbb{R}^d} \left(f(x) + \frac{1}{m}\sum_{j=1}^{m} B(g_j(x), \delta) \right).$$

Let $x^*(\delta)$ denote the unique minimizer of this relaxed problem. Our goal is to verify convergence to $x^*(\delta_\infty)$, which serves as an approximation of the true constrained solution x^*.

2.2 Proposed method: Tailed trajectory averaging

To address the high-variance and suboptimal convergence rate of the baseline method [4], we introduce a novel modification inspired by the classical work on optimal-rate stochastic approximation [15]. We term our algorithm **TA-SGD**.

The core idea is to apply the Polyak–Ruppert averaging technique, which is known to achieve an optimal convergence rate [15], but only *after* the algorithm has completed an initial **"burn-in" period**.

Our investigation reveals that a naive application of averaging starting from $k = 0$ fails (see section 4). This is because the initial iterates x_k are in a rapid transient phase, far from the optimal solution. Including these distant iterates in the average "poisons" the estimate, pulling it away from the true minimizer and drastically slowing convergence.

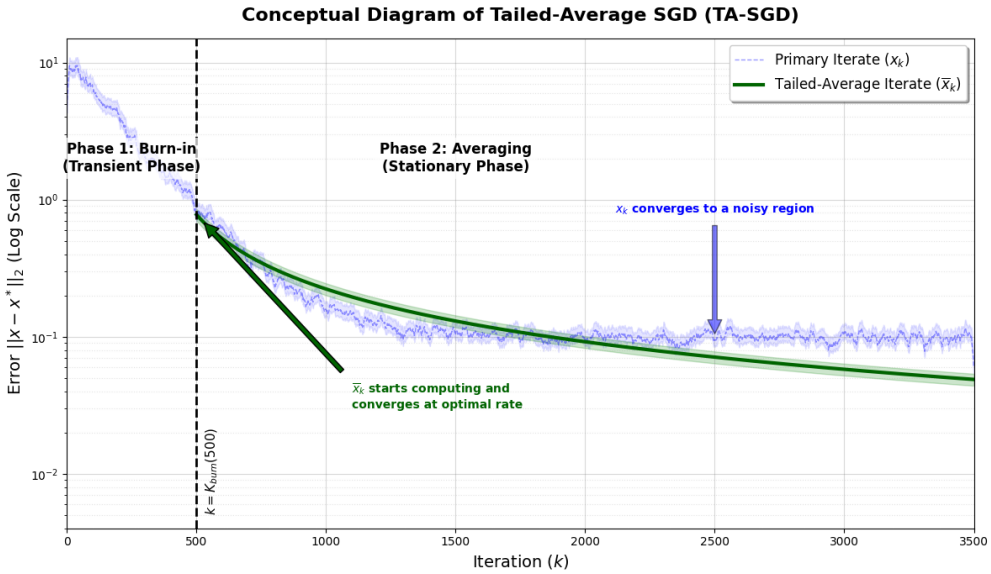

Figure 1: Conceptual diagram of the proposed TA-SGD mechanism. Phase 1 (Burn-in) allows the primary iterate x_k (blue, dashed) to reach a stationary region. Phase 2 (Averaging) begins at K_{burn}, computing $\overline{x}_k$ (green, solid), which converges at an optimal rate with low variance.

Our TA-SGD method avoids this by decoupling the optimization into two phases. First, a "burn-in" phase of K_{burn} iterations allows the primary iterate x_k to escape the transient

dynamics and settle into a neighborhood of the optimum $x^*(\delta_\infty)$. Second, an "averaging" phase begins, which systematically cancels the stochastic noise from sampling f_i and g_j.

This two-phase process is illustrated conceptually in Figure 1. The diagram shows how the primary iterate (x_k) converges to a high-variance stagnation region (Phase 1), while the tailed-average iterate ($\overline{x}_k$) only begins computation after the burn-in period (K_{burn}) and converges smoothly to the optimal solution with low variance (Phase 2).

This leads to our proposed TA-SGD update rule, which is defined by two sequences: the primary iterate x_k and the averaged iterate $\overline{x}_k$.

1. Primary iterate update (from [4]): The primary iterate x_k follows the original Adaptive Relaxed Barrier SGD update:

$$x_{k+1} = x_k - \gamma_k \left(\nabla f_{i_k}(x_k) + \nabla B(g_{j_k}(x_k), \delta_k) \right). \tag{2}$$

2. Tailed-average iterate update (our contribution): The final solution $\overline{x}_k$ is computed as a conditional average, which is only active after the burn-in period K_{burn}:

$$\overline{x}_{k+1} = \begin{cases} x_{k+1} & \text{if } k < K_{burn}, \\ \left(1 - \frac{1}{k - K_{burn}+1}\right) \overline{x}_k + \frac{1}{k-K_{burn}+1} x_{k+1} & \text{if } k \geq K_{burn}. \end{cases} \tag{3}$$

The final output of the algorithm is the averaged sequence $\overline{x}_k$. By waiting for the transient phase to complete, this update rule ensures that only high-quality iterates contribute to the final average, thereby achieving a low-variance solution and an optimal theoretical convergence rate.

2.3 Adaptive strategy for K_{burn} selection

A key practical question is how to determine the length of the burn-in period K_{burn}. While adaptive schemes for detecting stationarity exist in the literature [14], we provide a theoretical heuristic based on the nonasymptotic analysis of Polyak–Ruppert averaging.

According to [12], the error in SGD consists of two terms: a bias term related to the initial condition x_0, and a variance term related to the stochastic noise. The bias term decays geometrically at a rate of approximately $(1 - \gamma\theta)^k$ during the initial phase. To ensure that the iterates have "forgotten" the starting point and entered the steady-state regime (where averaging is most effective), the bias should be reduced to a small tolerance ϵ_{init}. This suggests that K_{burn} should satisfy

$$(1-\gamma\theta)^{K_{burn}} \leq \epsilon_{init} \implies K_{burn} \approx \frac{1}{\gamma\theta}\log\left(\frac{1}{\epsilon_{init}}\right).$$

Recognizing that the convergence speed is dictated by the condition number $\kappa = L/\theta$, this leads to the heuristic:

$$K_{burn} = \mathcal{O}\left(\kappa \log\left(\frac{1}{\epsilon_{init}}\right)\right) = \mathcal{O}\left(\frac{L}{\theta}\log\left(\frac{1}{\epsilon_{init}}\right)\right).$$

Remark on adaptivity: This formula defines an *adaptive mechanism* where the burn-in period scales automatically with the problem's condition number $\kappa = L/\theta$. Instead of being set as an arbitrary fixed hyperparameter, K_{burn} adapts to the optimization landscape's geometry, ensuring the averaging phase begins only after the specific transient dynamics of the problem have subsided.

For a detailed mathematical derivation of this heuristic, we refer the reader to Appendix C.

2.4 TA-SGD algorithm specification

The complete procedure for our proposed **TA-SGD** algorithm is detailed in Algorithm 1. The logic follows the two-phase process defined in Section 2.2, combining the primary iterate update (2) with the conditional averaging update (3).

3 Convergence analysis

In this section, we provide the formal proof of almost sure convergence for the proposed **TA-SGD** algorithm (Algorithm 1). This proof establishes that the algorithm's output converges to the correct solution. The analysis of its "rate" of convergence (i.e., its speed and superiority) is deferred to Section 3.2.

Remark on convergence analysis and novelty:

It is important to clarify the theoretical role of the burn-in period K_{burn}. Since K_{burn} is finite, it does not alter the *asymptotic* convergence rate of the averaging scheme; our asymptotic analysis in Section 3.2 naturally aligns with the classical framework of Polyak and Juditsky [15]. However, the novelty of TA-SGD lies in the **finite-time performance**. The standard theory assumes asymptotic behavior ($k \to \infty$), but in practice, initial transient iterates can significantly bias the average for finite K. The proposed modification ensures that the averaging mechanism is applied

Algorithm 1 TA-SGD for constrained optimization

Require: Initial point $x_0 \in \mathbb{R}^d$
Require: Number of iterations K
Require: Burn-in period $K_{burn} \geq 0$
Require: Step-size schedule $\{\gamma_k\}_{k=0}^{K-1}$
Require: Barrier-parameter schedule $\{\delta_k\}_{k=0}^{K-1}$
Initialize: Set averaged iterate $\overline{x}_0 \leftarrow x_0$
for $k = 0$ to $K-1$ **do**
  ▷ Phase 1: Primary Iterate Update (from [4])
  Sample an objective index $i_k \sim \mathcal{U}\{1, \ldots, n\}$
  Sample a constraint index $j_k \sim \mathcal{U}\{1, \ldots, m\}$
  Compute stochastic objective gradient: $G_f \leftarrow \nabla f_{i_k}(x_k)$
  Compute stochastic barrier gradient: $G_b \leftarrow \nabla B(g_{j_k}(x_k), \delta_k)$
  Update primary iterate: $x_{k+1} \leftarrow x_k - \gamma_k(G_f + G_b)$
  ▷ (2)
  ▷ Phase 2: Tailed-Average Iterate Update (Our Contribution)
  if $k < K_{burn}$ **then**
    ▷ Follow primary iterate during burn-in
    $\overline{x}_{k+1} \leftarrow x_{k+1}$
  else
    ▷ Set local averaging counter
    $N \leftarrow k - K_{burn} + 1$
    ▷ Compute recursive average
    $\overline{x}_{k+1} \leftarrow (1 - \frac{1}{N})\overline{x}_k + \frac{1}{N}x_{k+1}$
  end if
end for
▷ The final solution is the averaged iterate
return $\overline{x}_K$

only to the stationary distribution of the iterates, thereby bridging the gap between theoretical asymptotic rates and practical finite-time execution.

3.1 Almost sure convergence

Our proof builds directly upon the convergence guarantees established for the primary iterate sequence in the foundational work by Dimitrieski, Cao, and Ebenbauer [4], which itself relies on fundamental convergence theorems for stochastic processes [17].

Theorem 1 (Almost sure convergence of TA-SGD)**.** Consider the constrained optimization problem (1) and assume Assumptions 1, 2, and 3 hold. Let $\{\gamma_k\}_{k\geq 0}$ and $\{\epsilon_k\}_{k\geq 0}$ be positive sequences such that $\delta_k = \delta_\infty + \epsilon_k$ (with $\delta_\infty > 0$) satisfying

(a) $\sum_{k=0}^{\infty} \gamma_k = \infty$,

(b) $\sum_{k=0}^{\infty} \gamma_k^2 < \infty$,

(c) $\sum_{k=0}^{\infty} \gamma_k \epsilon_k < \infty$.

Then, for any finite burn-in period $K_{burn} \geq 0$, the sequence of tailed-average iterates $\{\overline{x}_k\}_{k\geq 0}$ generated by Algorithm 1 converges almost surely (a.s.) to the unique minimizer of the relaxed problem, $x^*(\delta_\infty)$. That is,

$$\lim_{k\to\infty} \overline{x}_k = x^*(\delta_\infty) \quad \text{a.s.}$$

Proof. The proof relies on two main steps:

Step 1: Convergence of the primary iterate $\{x_k\}$

The primary iterate sequence $\{x_k\}$, defined by (2), is generated by the Adaptive Relaxed Barrier SGD algorithm [4]. The conditions (a), (b), and (c) listed in Theorem 1 are the exact conditions required by [4, Theorem 1].

Therefore, we can directly apply their result, which proves that under these assumptions, the primary iterate sequence $\{x_k\}$ converges almost surely to the unique solution $x^*(\delta_\infty)$; that is,

$$\lim_{k\to\infty} x_k = x^*(\delta_\infty) \quad \text{a.s.}$$

Step 2: Convergence of the Tailed-Average Iterate $\{\overline{x}_k\}$

The final output of our algorithm is the tailed-average sequence $\{\overline{x}_k\}$, defined by (3) . For all $k \geq K_{burn}$, this sequence is the arithmetic mean (or Cesàro mean) of the "tail" of the primary sequence:

$$\overline{x}_k = \frac{1}{k - K_{burn} + 1} \sum_{i=K_{burn}}^{k} x_i$$

(The recursive update in Algorithm 1 is mathematically equivalent to this summation).

We have established in Step 1 that the sequence $\{x_k\}$ converges almost surely to the limit $L = x^*(\delta_\infty)$. A fundamental result in analysis (the Cesàro mean theorem) states that if a sequence $\{x_k\}$ converges to a limit L, its arithmetic mean also converges to L [9, 20].

This property holds for a "tailed average" as well. The first K_{burn} terms are a finite set of elements excluded from the average. As $k \to \infty$, the influence of this finite set on the average becomes zero. The convergence of $\{\overline{x}_k\}$ is determined entirely by the asymptotic behavior of the infinite tail $\{x_k\}_{k\geq K_{burn}}$, which converges to $x^*(\delta_\infty)$.

Since $x_k \xrightarrow{a.s.} x^*(\delta_\infty)$, it follows that its tailed-average $\overline{x}_k$ must also converge almost surely to the same limit; that is,

$$\lim_{k\to\infty} \overline{x}_k = x^*(\delta_\infty) \quad \text{a.s.}$$

This completes the proof of convergence. □

3.2 Convergence rate analysis

The goal is to prove that the MSE of the tailed-average iterate $\overline{x}_K$ achieves the optimal $O(1/K)$ rate. Our analysis adapts the classical framework established by Polyak and Juditsky [15] to demonstrate this result for our algorithm.

Step 1: Proof setup and definitions

We begin by defining the core variables and assumptions required for the proof, which are consistent with the problem formulation.

Definition 1 (Error notation)**.** Let $x^* = x^*(\delta_\infty)$ be the unique minimizer.

- $r_k = x_k - x^*$ is the error of the "primary iterate".
- $\overline{r}_K = \overline{x}_K - x^*$ is the error of the "tailed-average iterate".

Definition 2 (Update rule)**.** The primary iterate update (2) can be written in terms of the true expected gradient $g(x_k)$ and a noise term η_k:

$$x_{k+1} = x_k - \gamma_k \left(g(x_k) + \eta_k\right),$$

where $g(x_k) = \mathbb{E}[\nabla f_{i_k}(x_k) + \nabla B(g_{j_k}(x_k), \delta_k)|\mathcal{F}_{k-1}]$ is the true (expected) gradient of the relaxed objective and η_k is the noise. The noise term η_k is explicitly defined as the difference between the sampled stochastic gradient and the true expected gradient:

$$\eta_k = (\nabla f_{i_k}(x_k) + \nabla B(g_{j_k}(x_k), \delta_k)) - \mathbb{E}[\nabla f_{i_k}(x_k) + \nabla B(g_{j_k}(x_k), \delta_k)|\mathcal{F}_{k-1}].$$

Assumption 4 (Core assumptions)**.** This proof relies on the following key assumptions:

(A) **Primary convergence:** From Theorem 1, we know the primary iterate converges almost surely: $x_k \to x^*$ (or $r_k \to 0$).

(B) **Local linearity:** Since the objective is $\mathcal{C}^1$ (Assumption 1), the gradient $g(x_k)$ admits a linear approximation near the optimum x^*. This decomposition, which is a standard application of Taylor's theorem, separates the gradient into its linear approximation Hr_k and a nonlinear residual $\Delta(r_k)$:

$$g(x_k) = Hr_k + \Delta(r_k).$$

The critical part of this assumption is that the residual $\Delta(r_k)$ is "small", satisfying the condition $\|\Delta(r_k)\| \leq K_1\|r_k\|^{1+\lambda}$ for some $\lambda > 0$.[⋆]

(C) **Step-size:** The learning rate $\gamma_k = \gamma k^{-\theta}$ must satisfy $\theta \in (0.5, 1)$. This is critical for ensuring the nonlinear terms vanish upon averaging (see Appendix B).

(D) **Noise:** The noise η_k is a martingale difference sequence with zero mean ($\mathbb{E}[\eta_k|\mathcal{F}_{k-1}] = 0$) and bounded variance ($\mathbb{E}[\|\eta_j\|^2] \leq \sigma^2 < \infty$).

Step 2: Asymptotic equivalence

This is the first crucial step. We show that the tailed-average of our actual (nonlinear) process behaves asymptotically like the average of a simplified linear process.

Definition 3 (Actual vs. linearized process)**.** By subtracting x^* from the update rule and applying Assumption (B), the **actual (nonlinear) error dynamic** is

$$\begin{aligned} r_{k+1} &= x_{k+1} - x^* \\ &= (x_k - x^*) - \gamma_k(g(x_k) + \eta_k) \\ &= r_k - \gamma_k(Hr_k + \Delta(r_k) + \eta_k) \\ &= (I - \gamma_k H)r_k - \gamma_k\eta_k - \gamma_k\Delta(r_k). \end{aligned}$$

We compare this to the **linearized process** r_k^L, which omits the nonlinear term $\Delta(r_k)$:

$$r_{k+1}^L = (I - \gamma_k H)r_k^L - \gamma_k\eta_k.$$

[⋆] This specific condition is not an arbitrary simplification; it is the cornerstone assumption used in the foundational analysis of stochastic approximation. This property is precisely what Polyak and Juditsky [15] rely upon (see their Assumption 3.2) to rigorously prove that the averaged nonlinear terms vanish asymptotically. By satisfying this condition, our analysis can justifiably show that our process $\overline{r}_K$ inherits the convergence rate of the simpler, linearized process $\overline{r}_K^L$.

Lemma 1 (Asymptotic equivalence)**.** The scaled difference between the averaged nonlinear process ($\overline{r}_K$) and the averaged linearized process ($\overline{r}_K^L$) vanishes as $K \to \infty$:

$$\sqrt{K}\left\|\overline{r}_K - \overline{r}_K^L\right\| \xrightarrow{K\to\infty} 0 \quad \text{(in probability)}.$$

Proof. The objective is to demonstrate that the scaled average of the nonlinear residual terms, $\frac{1}{\sqrt{K}}\sum_{k=K_{burn}}^{K}\gamma_k\Delta(r_k)$, vanishes as $K \to \infty$.

The high-level logic relies on the interplay of our core assumptions. First, the burn-in period K_{burn} ensures that averaging only begins after the primary iterate r_k has converged to a small neighborhood of the optimum (due to Assumption A). Second, in this neighborhood, the linear approximation is valid, and the residual term $\Delta(r_k)$ becomes vanishingly small (per Assumption B). Finally, the step-size condition $\theta \in (0.5, 1)$ (Assumption C) provides the precise technical guarantee that the sum of these diminishing terms, when scaled, converges to zero.

The complete and rigorous technical justification, which adapts the foundational analysis from [15], is deferred to Appendix B. □

Corollary 1. Because the two processes are asymptotically equivalent, their MSEs share the same convergence rate.

$$\mathbb{E}[\|\overline{r}_K\|^2] \sim \mathbb{E}[\left\|\overline{r}_K^L\right\|^2].$$

Therefore, it is sufficient to analyze the convergence rate of the simpler **linearized process** $\overline{r}_K^L$.

Step 3: Analysis of the linearized process

We now analyze the convergence rate of the linearized process, $\overline{r}_K^L$. This analysis fundamentally relies on the algebraic decomposition (Lemma 2) derived from the original work of Polyak and Juditsky [15].

In that seminal work, the analysis was performed on the full average of the process (equivalent to setting $K_{burn} = 0$). While this might appear to be a simplification, it is essential to clarify that this analysis remains perfectly valid for our TA-SGD algorithm, which utilizes a $K_{burn} > 0$.

The reason is that the burn-in period (K_{burn}) only excludes a *finite* number of initial terms from the average. To formalize this justification, let $\overline{r}_K^{\text{full}} = \frac{1}{K}\sum_{i=1}^{K} r_i^L$ be the full average (as analyzed in [15]) and $\overline{r}_K = \frac{1}{K-K_{burn}}\sum_{i=K_{burn}+1}^{K} r_i^L$ be our tailed average (using the linearized process r_i^L and simplifying indices for asymptotic analysis). The full average can be decomposed as:

$$\overline{r}_K^{\text{full}} = \frac{1}{K}\left(\sum_{i=1}^{K_{burn}} r_i^L\right) + \frac{K - K_{burn}}{K}\left(\overline{r}_K\right).$$

We are interested in the convergence rate, which comes from the scaled error $\sqrt{K}\bar{r}_K$. Scaling the above equation by $\sqrt{K}$ yields:

$$\sqrt{K}\bar{r}_K^{\text{full}} = \underbrace{\frac{1}{\sqrt{K}}\left(\sum_{i=1}^{K_{burn}} r_i^L\right)}_{\text{Term (A)}} + \underbrace{\left(\frac{K-K_{burn}}{K}\right)}_{\text{Term (B)}}\left(\sqrt{K}\bar{r}_K\right).$$

As the total number of iterations $K \to \infty$, Term (A) vanishes to zero. This is because it is a *finite* sum (since K_{burn} is a fixed constant) scaled by $1/\sqrt{K}$. Simultaneously, the ratio in Term (B) converges to 1.

$$\lim_{K\to\infty} \sqrt{K}\bar{r}_K^{\text{full}} = \left(\lim_{K\to\infty} \frac{1}{\sqrt{K}} \sum_{i=1}^{K_{burn}} r_i^L\right) + \left(\lim_{K\to\infty} \frac{K-K_{burn}}{K}\right)\left(\lim_{K\to\infty} \sqrt{K}\bar{r}_K\right),$$

$$\lim_{K\to\infty} \sqrt{K}\bar{r}_K^{\text{full}} = (0) + (1) \cdot \lim_{K\to\infty} \sqrt{K}\bar{r}_K.$$

This rigorously demonstrates that the asymptotic behavior of the scaled full average is identical to that of our scaled tailed average. This fully justifies our use of the algebraic decomposition from [15] and confirms that the assumption noted in Lemma 2 (of setting $K_{burn} = 0$ for simplicity) is a valid step for the purpose of asymptotic analysis.

We now focus exclusively on the rate of $\bar{r}_K^L$. We use a precise algebraic decomposition derived in Appendix A (based on [15, Lemma 2]).

Lemma 2 (Algebraic decomposition)**.** For $K > K_{burn}$, the scaled averaged error $\sqrt{K}\bar{r}_K^L$ (assuming $K_{burn} = 0$ for simplicity) can be decomposed into three terms:

$$\sqrt{K}\bar{r}_K^L = I_K^{(1)} + I_K^{(2)} + I_K^{(3)},$$

where

- $I_K^{(1)} = \frac{1}{\sqrt{K}\gamma_0}\theta_K r_0$ (Bias from initial condition r_0),
- $I_K^{(2)} = \frac{1}{\sqrt{K}}\sum_{j=1}^{K-1} H^{-1}\eta_j$ (Main variance term from noise),
- $I_K^{(3)} = \frac{1}{\sqrt{K}}\sum_{j=1}^{K-1} w_j^K \eta_j$ (Residual term from step-size change).

We now compute the mean-squared norm $\mathbb{E}[\left\|\sqrt{K}\bar{r}_K^L\right\|^2]$ by analyzing each term.

Analysis of Bias Terms ($I^{(1)}$ and $I^{(3)}$)

- **Term $I^{(1)}$:** r_0 is a fixed constant, and the matrix θ_K is bounded. Due to the $1/\sqrt{K}$ scaling, this term vanishes: $\mathbb{E}[\left\|I_K^{(1)}\right\|^2] \to 0$ as $K \to \infty$.

- **Term $I^{(3)}$:** The matrices w_j^K are "small on average" (i.e., $\frac{1}{K}\sum \left\|w_j^K\right\| \to 0$). This is a direct consequence of the step-size condition $\theta \in (0.5, 1)$. This ensures the entire term vanishes in mean square: $\mathbb{E}[\left\|I_K^{(3)}\right\|^2] \to 0$.

Analysis of Variance Term ($I^{(2)}$)

This is the only term that remains and thus dictates the convergence rate.

$$\begin{aligned}
\mathbb{E}\left[\left\|I_K^{(2)}\right\|^2\right] &= \mathbb{E}\left[\left\|\frac{1}{\sqrt{K}}\sum_{j=1}^{K-1} H^{-1}\eta_j\right\|^2\right] \\
&= \frac{1}{K}\mathbb{E}\left[\left(\sum_{i=1}^{K-1}\eta_i^T(H^{-1})^T\right)\left(\sum_{j=1}^{K-1} H^{-1}\eta_j\right)\right] \\
&= \frac{1}{K}\sum_{i=1}^{K-1}\sum_{j=1}^{K-1}\mathbb{E}\left[\eta_i^T(H^{-1})^T H^{-1}\eta_j\right].
\end{aligned}$$

We now use the Martingale difference property (Assumption D). Since $\mathbb{E}[\eta_i|\mathcal{F}_{i-1}] = 0$, the expectations of all cross-terms are zero:

$$\mathbb{E}[\eta_i^T(H^{-1})^T H^{-1}\eta_j] = 0 \quad \text{for } i \neq j.$$

This simplifies the double summation to a single sum (assuming $H = H^T$):

$$\begin{aligned}
\mathbb{E}\left[\left\|I_K^{(2)}\right\|^2\right] &= \frac{1}{K}\sum_{j=1}^{K-1}\mathbb{E}\left[\eta_j^T H^{-2}\eta_j\right] \\
&\leq \frac{1}{K}\sum_{j=1}^{K-1}\mathbb{E}\left[\left\|H^{-1}\right\|_2^2\cdot\left\|\eta_j\right\|^2\right] \quad \text{(by matrix norm property)} \\
&\leq \frac{1}{K}\sum_{j=1}^{K-1}\left\|H^{-1}\right\|_2^2\cdot\sigma^2 \quad \text{(by Assumption D)} \\
&= \frac{K-1}{K}\left(\sigma^2\left\|H^{-1}\right\|_2^2\right).
\end{aligned}$$

As $K \to \infty$, we have $\frac{K-1}{K} \to 1$. Thus, the mean-squared norm of the dominant term converges to a constant:

$$\lim_{K\to\infty} \mathbb{E}\left[\left\|I_K^{(2)}\right\|^2\right] = \sigma^2 \left\|H^{-1}\right\|_2^2 = O(1).$$

Step 4: Final result

We now combine the results from the previous steps.

1. From Step 3, the scaled linearized error converges to a constant (is $O(1)$) because the bias terms vanish:

$$\mathbb{E}\left[\left\|\sqrt{K}\overline{r}_K^L\right\|^2\right] = \mathbb{E}\left[\left\|I_K^{(1)} + I_K^{(2)} + I_K^{(3)}\right\|^2\right] \xrightarrow{K\to\infty} \mathbb{E}\left[\left\|I_K^{(2)}\right\|^2\right] \leq O(1).$$

2. By dividing both sides by K, we get the rate for the linearized process:

$$\mathbb{E}\left[\left\|\overline{r}_K^L\right\|^2\right] \leq O(1/K).$$

3. From Step 2 (Asymptotic Equivalence), we know that the actual nonlinear process $\overline{r}_K$ inherits this same asymptotic rate.

This completes the proof, demonstrating that Tailed-Averaging successfully achieves the optimal $O(1/K)$ convergence rate.

$$\mathbb{E}\left[\left\|\overline{x}_K - x^*\right\|^2\right] \leq O(1/K).$$

4 Numerical experiments

In this section, we present a comprehensive empirical investigation designed to validate the theoretical claims of our proposed **TA-SGD** algorithm and demonstrate its practical advantages over existing methods. Our primary objectives are:

1. To empirically verify the optimal $O(1/K)$ convergence rate predicted by our analysis (Section 3.2).

2. To demonstrate the superiority of TA-SGD compared to the baseline Adaptive Relaxed Barrier SGD [4] by showing how it overcomes the convergence stagnation [8].

3. To confirm the detrimental effect of naive averaging (starting from $k = 0$) thereby justifying the necessity of the "burn-in" phase in TA-SGD.

4.1 Experimental setup

To ensure a fair and comprehensive evaluation, we followed a meticulous methodology in our experimental setup. The experiments were conducted on the Google Colab platform with the following specifications:

- **CPU:** Intel(R) Xeon(R) CPU @ 2.00GHz
- **RAM:** 13.61 GB
- **GPU:** NVIDIA Tesla T4 (15 GB VRAM)
- **OS:** Linux (6.6.97+)
- **Python Version:** 3.12.11

The experiments were performed on the academic quadratic programming (QP) problem introduced by Dimitrieski, Cao, and Ebenbauer [4], which is designed to test an algorithm's ability to handle large-scale optimization problems. The problem parameters were set as follows:

- Dimension: $d = 50$.
- Number of objective components: $n = 10$.
- Number of constraints: $m = 10,000$.

The objective function is defined as:

$$f(x) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{d}(\theta_{i,j}x_j + \log(1 + e^{-\theta_{i,j}x_j}) + (x_j - \gamma)^2).$$

To deeply analyze the algorithm's behavior, we adopted a Far Start Point ($x_0 = [10, \ldots, 10]^T$) for all our experiments. This challenging scenario, where the algorithm is initialized at a point known to be highly infeasible, serves as a true test of global stability and the ability to recover effectively.

4.1.1 Algorithms Compared

We compared the performance of three algorithms:

1. **Base SGD (Baseline):** The Adaptive Relaxed Barrier SGD algorithm from Dimitrieski, Cao, and Ebenbauer [4]. We used their primary parameter configuration: step-size $\gamma_k = 0.3k^{-0.8}$ and barrier decay $\epsilon_k = 5k^{-0.3}$ with $\delta_\infty = 10^{-6}$. This configuration was chosen as it represents a stable and effective baseline.

2. **Full Averaging (Naive):** The baseline SGD algorithm combined with Polyak–Ruppert averaging applied from the very first iteration ($k = 0$). This serves to demonstrate the "poisoning" effect discussed in Section 2.2.

3. **TA-SGD (Ours):** Our proposed TA-SGD algorithm (Algorithm 1). It uses the same primary update rule and parameters as the Base SGD, but activates Polyak–Ruppert averaging only after a burn-in period of $K_{burn} = 500$ iterations.

4.1.2 Performance Metric

Convergence was measured using the Euclidean norm of the error relative to the pre-computed optimum: $||x_k - x^*||_2$ (or $||\overline{x}_k - x^*||_2$ for averaging methods after the burn-in).

4.2 Performance evaluation

Figure 2 presents the core results of our comparative analysis, plotting the average error ($\pm$ one standard deviation) over `NUM_RUNS = 1000` independent runs for each algorithm.

The results clearly validate our theoretical expectations and demonstrate the effectiveness of TA-SGD:

- **Base SGD (Blue):** As predicted by theory [8] and observed in [4], the baseline algorithm converges rapidly initially but then stagnates, settling into a "stagnation region" around the optimum.

- **Full Averaging (Magenta)**: The naive application of averaging from $k = 0$ performs poorly. It converges much slower than the baseline during the initial phase, confirming the "poisoning effect" where early, inaccurate iterates corrupt the average. While it eventually surpasses the baseline's noise floor, its overall convergence is significantly hampered.

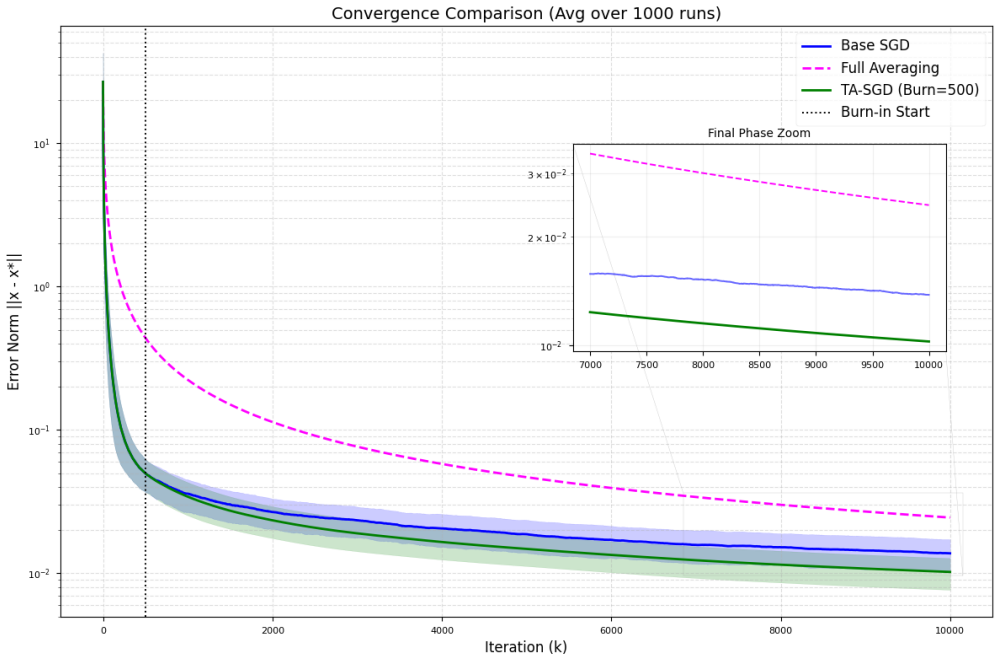

Figure 2: Algorithm Performance Comparison (Mean ± Std Dev over 1000 runs). The plot shows the convergence behavior of Base SGD [4], naive Full Averaging, and our proposed TA-SGD. The inset provides a zoomed view of the long-term behavior after the burn-in period ($K_{burn} = 500$).

- **TA-SGD (Green):** Our proposed algorithm exhibits the desired two-phase behavior:
  - **Phase 1 ($k < K_{burn}$):** TA-SGD tracks the error of the primary iterate x_k, mirroring the fast initial convergence of the baseline algorithm.
  - **Phase 2 ($k \geq K_{burn}$):** At iteration K_{burn} (marked by the vertical dashed line), averaging begins. The error curve for TA-SGD (now tracking $\overline{x}_k$) shows a distinct and sustained decrease, breaking away from the baseline's "stagnation region". The convergence in this phase appears consistent with the optimal $O(1/K)$ rate (approximately linear on the log-linear plot), achieving significantly higher precision than the baseline within the same number of iterations. The variance (shaded area) is also visibly reduced compared to the baseline in the later stages.

These results strongly support our central hypothesis: the burn-in phase is crucial for avoiding the pitfalls of naive averaging, and the subsequent tailed-averaging effectively overcomes the noise floor limitation of the baseline SGD, leading to faster convergence and higher accuracy. It is worth noting that the additional computational overhead of the recursive averaging update (3) is negligible compared to the cost of computing the stochastic gradients (2) in each itera-

tion. Consequently, the iteration-based performance comparison shown in Figure 2 serves as an accurate proxy for the practical wall-clock time efficiency of the algorithms.

While Figure 2 illustrates the dynamic convergence behavior, **Table 1** provides a concise numerical summary of the final results. The table quantifies the outcome at $K = 10,000$ iterations, confirming that our proposed TA-SGD algorithm achieves the lowest mean final error (1.02×10^{-2}). This result is a clear improvement over the Base SGD baseline (1.38×10^{-2}) and numerically validates our central hypothesis: the burn-in phase successfully mitigates the "poisoning effect" of early iterates, and the subsequent tailed averaging effectively overcomes the stagnation.

Table 1: Comprehensive Performance Statistics (Mean ± Std Dev) over $R = 1000$ independent runs. The proposed TA-SGD uses a burn-in period of $K_{burn} = 500$ based on the adaptive heuristic.

Metric	Base SGD	Full Averaging	TA-SGD (Ours)
Final Error ($\|\|x_K - x^*\|\|$)	$1.38 \times 10^{-2}(\pm 3.48 \times 10^{-3})$	$2.45 \times 10^{-2}(\pm 1.41 \times 10^{-2})$	$1.02 \times 10^{-2}(\pm 2.55 \times 10^{-3})$
Objective Value ($f(x)$)	81.7414 ± 0.0001	81.7420 ± 0.0021	81.7413 ± 0.0001
Max Constraint Violation	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Avg. CPU Time (s)	0.5924	0.5924	0.5924**

4.3 Sensitivity to burn-in period (K_{burn})

A key parameter in TA-SGD is the burn-in period, K_{burn}. To assess the algorithm's robustness, we performed a sensitivity analysis by varying K_{burn} while keeping other parameters fixed.

Figure 3 illustrates the sensitivity of the algorithm to the burn-in period K_{burn}.

- **Insufficient burn-in ($K_{burn} < 500$):** The performance degrades significantly as $K_{burn} \to 0$, approaching the high error of the naive Full Averaging approach. This empirically confirms that early averaging without a sufficient burn-in phase is detrimental due to the "poisoning effect" of initial transient iterates.

- **Optimal region ($500 \leq K_{burn} \leq 3000$):**)The algorithm achieves its global minimum error at $K_{burn} = 500$. The performance remains stable and near-optimal throughout this range. The empirical results demonstrate that our adaptive selection strategy (derived in Section 2.3) correctly identifies the optimal switching point, confirming that the theoretical condition number analysis translates effectively to practice.

** The computational overhead of the scalar averaging update update (3) is negligible compared to the gradient computation. Consequently, the wall-clock time is statistically identical for all three algorithms.

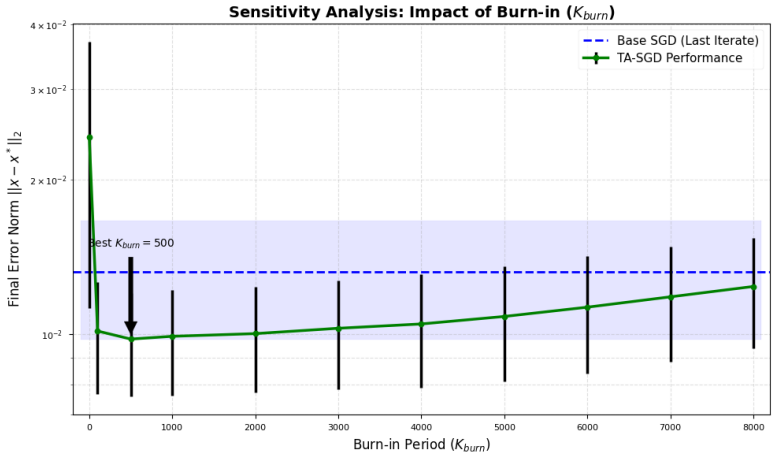

Figure 3: Sensitivity analysis of TA-SGD with respect to the burn-in period (K_{burn}). The plot shows the final error achieved after 10,000 iterations for various K_{burn} values. Although the error at $K_{burn} = 0$ (approximating Full Averaging) appears close to the optimum on this scale, the difference corresponds to a significantly higher variance and final error as detailed in Table 1.

- **Excessive burn-in ($K_{burn} > 4000$):** The error increases slightly but remains robustly lower than the Base SGD baseline (as shown by the blue dashed line). This gradual increase occurs because delaying the start of the averaging phase reduces the effective sample size ($N = K - K_{burn}$) available for noise cancellation within the fixed budget of $K = 10,000$ iterations.

This sensitivity analysis underscores the robustness of the proposed **adaptive selection strategy**. While precise manual tuning is not strictly necessary due to the wide optimal region, the theoretical heuristic allows us to automatically select an effective burn-in period (e.g., $K_{burn} = 500$) that ensures the primary iterate has entered the stationary phase before averaging begins.

5 Conclusion

In this paper, we introduced TA-SGD, a novel and efficient modification for solving stochastic constrained optimization problems. Our work builds upon the modern framework established by Dimitrieski, Cao, and Ebenbauer [4], which utilizes adaptive relaxed barrier functions to handle constraints.

We identified a critical limitation in this baseline approach: while it guarantees convergence, it suffers from convergence stagnation, settling into a high-variance "stagnation region" around

the optimum. This limitation prevents the algorithm from achieving the optimal convergence rate.

To overcome this, we drew inspiration from the seminal work of Polyak and Juditsky [15], on stochastic approximation. We proposed integrating their classical averaging technique, which is known to effectively cancel stochastic noise and accelerate convergence.

Our primary contribution is the "tailed-average" mechanism. We demonstrated, both theoretically and empirically, that a naive application of averaging from the first iteration fails, as the estimate is "poisoned" by the initial, high-error transient iterates. Our TA-SGD algorithm intelligently circumvents this by introducing a "burn-in" period (K_{burn}), allowing the primary iterate to first settle into a stationary region. Only then does averaging begin, ensuring that only high-quality iterates contribute to the final solution.

Our rigorous theoretical analysis established that this two-phase approach achieves the best of both worlds:

1. It preserves the strong almost sure convergence guarantees of the underlying relaxed barrier method [4].

2. It successfully breaks the noise floor to achieve the optimal $O(1/K)$ convergence rate established by Polyak and Juditsky [15].

Our numerical experiments provided a clear validation of these theoretical claims. The results demonstrated that TA-SGD decisively overcomes the stagnation of the baseline algorithm, achieving significantly faster convergence and a final solution of substantially higher precision.

Future Work

The promising results of our TA-SGD algorithm open several interesting avenues for future research. Based on our findings, we suggest the following directions:

- **Theoretical Extensions to General Convexity:** Our convergence rate analysis successfully established the optimal $O(1/K)$ rate, but it relies on the standard assumption of θ-strong convexity. A valuable theoretical extension would be to analyze the convergence rate of TA-SGD for general convex (i.e., nonstrongly convex) objective functions, following recent advances in analyzing SGD for nonconvex functions satisfying weaker conditions such as the Polyak-Lojasiewicz property [10]

- **Synergy with Nonuniform Sampling:** Our algorithm currently employs standard uniform sampling for both objectives and constraints. Future work could explore the synergy between Tailed-Averaging and nonuniform or importance sampling techniques. For instance, adaptively sampling constraints that are more frequently violated might accelerate the burn-in phase, allowing the algorithm to reach the stationary region (and thus begin averaging) more quickly.

A Justification of linear decomposition (Step 3)

This appendix justifies the decomposition $\sqrt{K}\overline{r}_K^L = I^{(1)} + I^{(2)} + I^{(3)}$ used in Step 3. This derivation follows the algebraic steps of Lemma 2 in the appendix of [15], which results in the precise decomposition shown in Equation (A10) of that work. The linearized process is $r_k^L = (I - \gamma_{k-1}H)r_{k-1}^L - \gamma_{k-1}\eta_{k-1}$. By unrolling this recurrence from r_0, we get the explicit solution:

$$r_k^L = X_0^k r_0 - \sum_{j=0}^{k-1} X_{j+1}^k \gamma_j \eta_j,$$

where $X_j^k = \prod_{i=j}^{k-1}(I - \gamma_i H)$ is the state transition matrix.

The average $\overline{r}_K^L$ is $\frac{1}{K}\sum_{k=0}^{K-1} r_k^L$:

$$\overline{r}_K^L = \left(\frac{1}{K}\sum_{k=0}^{K-1} X_0^k\right) r_0 - \frac{1}{K}\sum_{k=0}^{K-1}\sum_{j=0}^{k-1} X_{j+1}^k \gamma_j \eta_j.$$

We apply summation by parts (changing the order of the double summation):

$$\overline{r}_K^L = \left(\frac{1}{K}\sum_{k=0}^{K-1} X_0^k\right) r_0 - \frac{1}{K}\sum_{j=0}^{K-1}\left(\sum_{k=j+1}^{K-1} X_{j+1}^k \gamma_j\right) \eta_j.$$

The core insight (from [15]) is that the inner sum (the linear accumulation) can be approximated by H^{-1}.[***]. We define the residual w_j^K:

$$\sum_{k=j+1}^{K-1} X_{j+1}^k \gamma_j = H^{-1} - w_j^K.$$

[***] This approximation is derived from the geometric series sum. In the idealized case where the step-size is constant ($\gamma_k = \gamma$), the inner sum $\sum_{k=j+1}^{K-1} X_{j+1}^k \gamma_j$ simplifies to $\gamma\sum_{n=0}^{K-j-2}(I - \gamma H)^n$. As $K \to \infty$, this sum converges to $\gamma[I - (I - \gamma H)]^{-1} = \gamma[\gamma H]^{-1} = H^{-1}$. The term w_j^K defined next thus captures the residual error from this ideal approximation, which arises because the step-sizes γ_k are not constant.

Substituting this back and defining $\theta_K = \gamma_0 \sum X_0^k$, we have

$$\begin{aligned}\overline{r}_K^L &= \left(\frac{1}{\gamma_0 K}\theta_K\right) r_0 - \frac{1}{K}\sum_{j=0}^{K-1}(H^{-1} - w_j^K)\eta_j \\ &= \frac{1}{\gamma_0 K}\theta_K r_0 - \frac{1}{K}\sum_{j=0}^{K-1} H^{-1}\eta_j + \frac{1}{K}\sum_{j=0}^{K-1} w_j^K \eta_j.\end{aligned}$$

Multiplying the entire equation by $\sqrt{K}$ yields the desired decomposition:

$$\sqrt{K}\overline{r}_K^L = \underbrace{\frac{1}{\sqrt{K}\gamma_0}\theta_K r_0}_{I^{(1)}} + \underbrace{\frac{1}{\sqrt{K}}\sum_{j=0}^{K-1} H^{-1}\eta_j}_{I^{(2)}} + \underbrace{\frac{1}{\sqrt{K}}\sum_{j=0}^{K-1} w_j^K \eta_j}_{I^{(3)}}.$$

The condition $\theta \in (0.5, 1)$ (Assumption C) is crucial to prove that $\frac{1}{K}\sum ||w_j^K|| \to 0$, which ensures $I^{(3)}$ vanishes.[†]

B Justification of asymptotic equivalence (Step 2)

This appendix justifies why the nonlinear terms $\Delta(r_k)$ can be ignored when averaging. We need to show that $\sqrt{K}\left\|\overline{r}_K - \overline{r}_K^L\right\| \to 0$.

This difference is proportional to the scaled average of the nonlinear residuals:

$$\text{Difference} \propto \frac{1}{\sqrt{K}}\sum_{k=K_{burn}}^{K} \gamma_k \Delta(r_k).$$

To prove this scaled average vanishes, we apply a standard result from stochastic approximation (see [15]), which is a direct consequence of Kronecker's Lemma. This lemma effectively transforms our original problem (showing the scaled average converges to zero) into an equivalent, simpler task: proving that a related sum is finite. Using Assumption (B), $||\Delta(r_k)|| \leq K_1||r_k||^{1+\lambda}$, this task reduces to proving that the following sum S converges:

$$S = \sum_{k=1}^{\infty} \frac{||r_k||^{1+\lambda}}{\sqrt{k}} < \infty.$$

[†] This specific range represents a critical balance. If $\theta \leq 0.5$, the step-size γ_k decays too slowly, causing the approximation error w_j^K to be too large on average for its mean to vanish. Conversely, if $\theta \geq 1$, γ_k decays so quickly that its sum becomes finite ($\sum \gamma_k < \infty$), which violates the primary convergence condition (a) of Theorem 1. This "sweet spot" ensures the error term $I^{(3)}$ vanishes while still guaranteeing convergence.

To prove S converges, we test the convergence of its expectation, $\mathbb{E}[S]$. To do this, we first need the convergence rate of the primary (nonaveraged) iterate r_k.

While the formal proof requires a detailed Lyapunov analysis (as shown in [15]), this $O(k^{-\theta})$ rate is the expected standard convergence rate for an SGD iterate that uses a step-size of $\gamma_k \propto k^{-\theta}$.

From the analysis in [15], we can use this established bound:

$$\mathbb{E}[||r_k||^2] \propto \mathbb{E}[V_k] \leq K_c\gamma_k \leq K_c k^{-\theta} \quad \text{(for large k)}.$$

We apply Jensen's Inequality using the function $f(x) = x^{(1+\lambda)/2}$. This function is convex for $x \geq 0$ because the exponent $p = (1+\lambda)/2 \geq 1$ (since $\lambda \geq 1$). Applying the inequality yields:

$$\mathbb{E}[\|r_k\|^{1+\lambda}] = \mathbb{E}\left[(\|r_k\|^2)^{(1+\lambda)/2}\right] \leq \left(\mathbb{E}[\|r_k\|^2]\right)^{(1+\lambda)/2}$$

$$\mathbb{E}[\|r_k\|^{1+\lambda}] \leq (K_c\gamma k^{-\theta})^{(1+\lambda)/2}.$$

Now, we test the convergence of $\mathbb{E}[S]$:

$$\begin{aligned}\mathbb{E}[S] &= \sum_{k=1}^{\infty} \frac{\mathbb{E}[\|r_k\|^{1+\lambda}]}{\sqrt{k}} \leq \sum_{k=1}^{\infty} \frac{(K_c\gamma k^{-\theta})^{(1+\lambda)/2}}{k^{1/2}} \\ &= C\sum_{k=1}^{\infty} \frac{k^{-\theta(1+\lambda)/2}}{k^{1/2}} \\ &= C\sum_{k=1}^{\infty} k^{-\left(\frac{\theta(1+\lambda)}{2}+\frac{1}{2}\right)}.\end{aligned}$$

This is a p-series, which converges if and only if the exponent $p > 1$:

$$p = \frac{\theta(1+\lambda)}{2} + \frac{1}{2} > 1,$$

$$\frac{\theta(1+\lambda)}{2} > \frac{1}{2} \implies \theta(1+\lambda) > 1.$$

This condition, $\theta(1+\lambda) > 1$, is not arbitrary; it is the exact requirement imposed by [15] in their Assumptions 3.4 and 4.7. As noted in their commentary following Assumption 3.4, for the standard $\mathcal{C}^2$ case where $\lambda = 1$, this implies $2\theta > 1$ or $\theta > 0.5$. This confirms that our Assumption (C) aligns with the conditions required by the original proof.

Since $\mathbb{E}[S] < \infty$, the sum S converges almost surely. By Kronecker's Lemma, the convergence of this sum implies that the scaled average vanishes. This rigorously justifies Step 2.

C Derivation of the adaptive burn-in heuristic

In this appendix, we provide the formal derivation for the heuristic estimate of K_{burn} presented in section 2.2. The goal is to determine the number of iterations k required for the influence of the initial condition x_0 to become negligible compared to the stochastic noise.

Under Assumption 1 (strong convexity with parameter θ and L-smoothness), the expected distance to the optimum at iteration k satisfies the standard recurrence relation [1]:

$$\mathbb{E}[\|x_k - x^*\|^2] \leq (1 - 2\gamma_k\theta)\mathbb{E}[\|x_{k-1} - x^*\|^2] + \gamma_k^2\sigma^2.$$

During the transient "burn-in" phase, the error is dominated by the initial distance $\|x_0 - x^*\|^2$ rather than the variance term σ^2. Focusing on the contraction of this initial bias with a constant step-size approximation γ, we have:

$$\|x_k - x^*\|^2 \approx (1 - \gamma\theta)^k\|x_0 - x^*\|^2.$$

We seek the smallest k such that the initial error is reduced to a tolerance ϵ_{init}:

$$(1 - \gamma\theta)^k \leq \epsilon_{init}.$$

Taking the natural logarithm of both sides:

$$k\ln(1 - \gamma\theta) \leq \ln(\epsilon_{init}).$$

Using the Taylor expansion $\ln(1 - x) \approx -x$ for small $x = \gamma\theta$, we obtain:

$$k(-\gamma\theta) \lesssim -\ln\left(\frac{1}{\epsilon_{init}}\right) \implies k \gtrsim \frac{1}{\gamma\theta}\ln\left(\frac{1}{\epsilon_{init}}\right).$$

For stability in smooth optimization, the step size is typically bounded by $\gamma \leq \frac{1}{L}$. Substituting $\gamma \approx \frac{1}{L}$:

$$k \gtrsim \frac{L}{\theta}\ln\left(\frac{1}{\epsilon_{init}}\right).$$

Defining the condition number as $\kappa = L/\theta$, we recover the heuristic:

$$K_{burn} = \mathcal{O}\left(\kappa\log\left(\frac{1}{\epsilon_{init}}\right)\right).$$

This confirms that the required length of the burn-in period scales linearly with the condition number of the problem.

D Experimental details

This appendix provides the specific implementation and parameter details used to generate the numerical results in section 4, ensuring full reproducibility. All experiments were conducted in Python using the `NumPy` library for numerical operations and `Matplotlib` for visualization.

D.1 Problem generation

The experiments were performed on the quadratic programming (QP) problem introduced by Dimitrieski, Cao, and Ebenbauer [4], as described in Section 4.1. The problem was generated once using a master seed (`master_seed = 42`) with the following parameters:

- Dimension: $d = 50$
- Objective components: $n = 10$
- Number of constraints: $m = 10,000$
- Objective parameter: $\gamma = 2.365$. [‡]

The objective function coefficients $\theta \in \mathbb{R}^{n\times d}$ were sampled from $\mathcal{U}(0.1, 1.1)$. The constraint gradients $A \in \mathbb{R}^{m\times d}$ were generated by first defining a diagonal matrix Q with entries sampled from $\mathcal{U}(1, 1.5)$, and then sampling m points on the surface of the ellipsoid defined by Q to form the constraint matrix A.

D.2 Algorithm implementation and hyperparameters

We implemented the three algorithms (Base SGD, Full Averaging, and TA-SGD) as described in Section 4.1.

[‡] Following the benchmark design in [4], these parameters are calibrated such that the unconstrained global minimizer $x_f^* = \arg\min f(x)$ satisfies $\|x_f^*\|_2 = 15$. This guarantees that $x_f^* \notin \mathcal{E}$, ensuring that the constraints are active at the optimal solution.

- **Total iterations:** All algorithms were run for $K = 10,000$ iterations.

- **Starting point:** All runs were initialized at the "Far Start Point" $x_0 = [10, \dots, 10]^T \in \mathbb{R}^{50}$.

- **Schedules:** The step-size and barrier parameter schedules followed the configuration in [4]:

$$\gamma_k = 0.3 \cdot k^{-0.8} \quad \text{and} \quad \epsilon_k = 5.0 \cdot k^{-0.3}$$

  with the final barrier parameter set to $\delta_\infty = 10^{-6}$.

D.3 Reproducibility and analysis

- **Reference solution (x^*):** The optimal solution $x^* = x^*(\delta_\infty)$ was pre-computed using a *deterministic full-gradient descent method*. Unlike the stochastic approximations, this solver utilized the full dataset at each iteration (computing the exact full gradient $\nabla f(x)$ and $\nabla B(x)$ without sampling). This deterministic process was executed for $K_{long} = 50,000$ iterations to ensure convergence to a high-precision ground truth, completely independent of the stochastic algorithms being evaluated.

- **Statistical analysis:** The results in Figure 2 and Table 1 were generated by averaging over $R = 1,000$ independent runs.

- **Seeding:** To ensure statistically independent yet reproducible trials, the main problem was generated with `master_seed = 42`, and each of the $R = 1,000$ runs (indexed by $r \in [0, \dots, 999]$) was seeded with `master_seed` $+ r + 2$.

- **Metrics:** The "Error Norm" (Figure 2) corresponds to the mean of $||x_k - x^*||_2$ (or $||\overline{x}_k - x^*||_2$) at each iteration k across the R runs. The shaded region represents ± 1 standard deviation. The "Mean Final Error" (Table 1) is the mean and standard deviation of this error norm specifically at the final iteration, $k = 10,000$.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Bottou, L., Curtis, F.E. and Nocedal, J. *Optimization methods for large-scale machine learning,* SIAM Rev. 60(2) (2018), 223–311.

[2] Boyd, S. and Vandenberghe, L. *Convex optimization*, Cambridge University Press, 2004.

[3] Bubeck, S. *Convex optimization: Algorithms and complexity,* Found. Trends Mach. Learn. 8(3-4) (2015), 231–358.

[4] Dimitrieski, N., Cao, J. and Ebenbauer, C. *Stochastic gradient descent for constrained optimization based on adaptive relaxed barrier functions,* Int. J. Robust. Nonlinear Control. (2025).

[5] Feller, C. and Ebenbauer, C. *A stabilizing iteration scheme for model predictive control based on relaxed barrier functions*, Automatica 80 (2017), 328–339.

[6] Fiacco, A.V. and McCormick, G.P. Nonlinear programming: Sequential unconstrained minimization techniques, John Wiley & Sons, 1968.

[7] Garrigos, G. and Gower, R.M. *Handbook of convergence theorems for (stochastic) gradient methods,* arXiv preprint arXiv:2301.11235 (2023).

[8] Gower, R.M., Loizou, N., Qian, X., Sailanbayev, A., Shulgin, E. and Richtárik, P. *SGD: General analysis and improved rates,* arXiv preprint arXiv:1901.09401 (2019).

[9] Hardy, G.H. *Divergent series*, Oxford University Press, 1949.

[10] Karandikar, R.L. and Vidyasagar, M. *Convergence rates for stochastic approximation: biased noise with unbounded variance, and applications,* arXiv preprint arXiv:2312.02828 (2024).

[11] Li, M., Grigas, P. and Atamtürk, A. *New penalized stochastic gradient methods for linearly constrained strongly convex optimization,* SIAM J. Optim. 32(2) (2022), 859–887.

[12] Moulines, E. and Bach, F. *Non-asymptotic analysis of stochastic approximation algorithms,* In Advances in Neural Information Processing Systems (NeurIPS), 451–459, 2011.

[13] Nocedal, J. and Wright, S.J. Numerical optimization, 2nd ed., Springer, 2006.

[14] Pflug, G.C. *Non-asymptotic confidence bounds for stochastic approximation algorithms with constant step size,* Monatsh. Math. 110(3-4) (1990), 297–314.

[15] Polyak, B.T. and Juditsky, A.B. *Acceleration of stochastic approximation by averaging,* SIAM J. Control Optim. 30(4) (1992), 838–855.

[16] Robbins, H. and Monro, S. *A stochastic approximation method*, Ann. Math. Stat. 22(3) (1951), 400–407.

[17] Robbins, H. and Siegmund, D. *A convergence theorem for nonnegative almost supermartingales and some applications,* In Optimizing Methods in Statistics, Academic Press, 1971, 233–257.

[18] Rockafellar, R.T. Convex analysis, Princeton University Press, 1970.

[19] Roy, S.K. and Harandi, M. *Constrained stochastic gradient descent: The good practice,* In Int. Conf. Digit. Image Comput. Tech. Appl. (DICTA), IEEE, 1–8, 2017.

[20] Rudin, W. *Principles of mathematical analysis*, 3rd ed., McGraw-Hill, 1976.

[21] Wang, M. and Bertsekas, D.P. *Incremental constraint projection methods for variational inequalities,* Math. Program. 150 (2015), 321–363.

[22] Wang, X., Ma, S. and Yuan, Y. *Penalty methods with stochastic approximation for stochastic nonlinear programming,* arXiv preprint arXiv:1605.05609 (2016).

[23] Yan, Y. and Xu, Y. *Adaptive primal-dual stochastic gradient method for expectation-constrained convex stochastic programs,* Math. Program. Comput. 14(2) (2022), 319–363.

[24] Zhang, T. *Solving large scale linear prediction problems using stochastic gradient descent algorithms,* In Proc. 21st Int. Conf. Mach. Learn. (ICML), 116, 2004.