ON THE CONVERGENCE OF STOCHASTIC LOW-RANK ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-rank adaptation (LoRA) minimizes the training loss after adding a trainable low-rank adapters to a frozen pretrained weight matrix. Previous analysis of gradient descent for LoRA (LoRA-GD) gives a sufficient complexity of $\exp(\mathcal{O}(\epsilon^{-2}))$ iterations to find an ϵ-stationary point in the deterministic setting. We sharpen this analysis and show that $\mathcal{O}(\epsilon^{-4})$ iterations suffice for the same first-order criterion. We further study stochastic LoRA under unbiased stochastic gradients with finite variance. We propose LoRA-NSGDM, which finds an ϵ-stationary point with $\mathcal{O}(\epsilon^{-8})$ stochastic oracle calls. Under an additional mean-square smoothness condition, we propose LoRA-STORM, which improves the stochastic oracle complexity to $\mathcal{O}(\epsilon^{-6})$.

1 INTRODUCTION

Low-rank adaptation (LoRA) reduces fine-tuning costs by learning a low-rank update to frozen pretrained weights (Hu et al., 2022). Given a loss $\mathcal{L} : \mathbb{R}^{m\times n} \to \mathbb{R}$ and frozen weights $W_{\text{base}} \in \mathbb{R}^{m\times n}$, we optimize factors $B \in \mathbb{R}^{m\times r}$ and $A \in \mathbb{R}^{r\times n}$ that parameterize the update as sBA, where s is a fixed scaling parameter. Defining $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ by $f(X) := \mathcal{L}(W_{\text{base}} + sX)$ and stacking the factors as $V = [B; A^\top] \in \mathbb{R}^{(m+n)\times r}$, we study the factorized problem

$$\min_{V\in\mathbb{R}^{(m+n)\times r}} J(V) \quad \text{where} \quad J(V) := f(BA). \tag{1}$$

Analyzing the convergence guarantee for LoRA (1) remains challenging. As observed in prior work (Sun et al., 2024; Malinovsky et al., 2024), even when the loss f has a Lipschitz gradient, the factorized objective J generally has no globally Lipschitz gradient in (B, A) because the map $(B, A) \mapsto BA$ is bilinear. Consequently, standard convergence theory for smooth nonconvex objectives does not directly apply to LoRA. Existing theory studies LoRA's expressive power (Zeng & Lee, 2024), optimization landscapes and convergence in the NTK regime and beyond (Jang et al., 2024; Kim et al., 2025), and gradient-flow dynamics in matrix factorization (Xu et al., 2025). There are also studies on factor-specific learning rates and gradient preconditioning in infinite-width limits (Hayou et al., 2024; Zhang & Pilanci, 2024), or impose structural conditions on the LoRA factors (Ding et al., 2026). These results do not establish a non-asymptotic convergence guarantee for LoRA gradient descent under only smoothness and lower boundedness of the original loss.

Recently, Mu & Klabjan (2026) provide a non-asymptotic analysis of LoRA-GD under smoothness and lower boundedness condition of the loss $\mathcal{L}$. They prove that LoRA-GD converges at a rate of

$$\min_{0\le t<T} \|\nabla J(V_t)\|_F^2 = \mathcal{O}\left(\frac{1}{\log T}\right).$$

This rate yields an exponential complexity bound $\exp(\mathcal{O}(\epsilon^{-2}))$ for finding an ϵ-stationary point, which is not satisfactory. This motivates us to ask

Q1. Can we improve the iteration complexity of LoRA-GD to a polynomial bound?

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: We compare iteration complexity in the deterministic setting and oracle complexity in the stochastic setting for finding an ϵ-stationary point of J (cf. Definition 2.1). MSS denotes mean-square smoothness.

Method	Setting	Complexity	Reference
LoRA-GD	Deterministic	$\exp(\mathcal{O}(\epsilon^{-2}))$	Mu & Klabjan (2026)
LoRA-GD	Deterministic	$\mathcal{O}(\epsilon^{-4})$	Theorem 3.1
LoRA-NSGDM	Finite variance	$\mathcal{O}(\epsilon^{-8})$	Theorem 4.5
LoRA-STORM	Finite variance + MSS	$\mathcal{O}(\epsilon^{-6})$	Theorem 5.2

Despite the non-asymptotic analysis for the deterministic setting, the stochastic counterpart remains open in Mu & Klabjan (2026), which identifies the control of a fourth-order noise moment as a central difficulty. In addition, the bounded variance assumption on the stochastic gradient of $f(\cdot)$ cannot lead to the bounded variance of the stochastic gradient of $J(\cdot)$. Existing stochastic guarantees either impose additional global regularity and trajectory conditions (Jiang et al., 2024) or alter the factor update (Sokolov et al., 2025). Neither establishes convergence of stochastic LoRA under finite variance condition on stochastic gradient of $f(\cdot)$ alone. This motivates our second question:

> *Q2. Can we develop convergent methods for stochastic LoRA under standard finite variance assumptions?*

We answer both questions affirmatively through the following contributions.

- In the deterministic setting, we sharpen the trajectory analysis of Mu & Klabjan (2026) and prove that $\mathcal{O}(\epsilon^{-4})$ iterations suffice to find V satisfying $\|\nabla J(V)\|_F \le \epsilon$ (Section 3).
- In the stochastic setting, we show that LoRA-SGD may fail to converge in expectation even when the stochastic gradients of f are unbiased and have bounded variance. To address this limitation, we develop LoRA-NSGDM, which combines joint normalization of the two factor directions with exponential moving-average momentum (Cutkosky & Mehta, 2020). Under smoothness and lower boundedness of f and the same oracle assumptions, we prove that LoRA-NSGDM returns a point V satisfying $\mathbb{E}\,\|\nabla J(V)\|_F \le \epsilon$ using $\mathcal{O}(\epsilon^{-8})$ stochastic oracle calls (Section 4).
- When the stochastic oracle additionally satisfies mean-square smoothness, we develop LoRA-STORM, which applies the recursive momentum Cutkosky & Orabona (2019) to the normalized stochastic gradient. We show an improved stochastic oracle complexity of $\mathcal{O}(\epsilon^{-6})$ to find the ϵ stationary point (Section 5).

Table 1 summarizes the our results with the prior works.

Organization. The remainder of the paper is organized as follows. Section 2 introduces the LoRA factorization, the stationarity measure, and the assumptions on the objective and stochastic oracle. Section 3 establishes the polynomial convergence bound for LoRA-GD. Section 4 explains the limitation of plain LoRA-SGD and develops LoRA-NSGDM under finite variance. Section 5 introduces LoRA-STORM under mean-square smoothness and establishes its improved stochastic oracle complexity. In Section 6, we provides empirical evaluations of the proposed methods. Section 7 concludes the paper. We leave the discussion of related work and the detailed proof in the Appendix.

2 Preliminaries

We fix integers $m, n, r \ge 1$ with $r \le \min\{m, n\}$. We use the Frobenius inner product and norm

$$\langle X, Y\rangle_F := \mathrm{tr}(X^\top Y) \quad \text{and} \quad \|X\|_F := \sqrt{\langle X, X\rangle_F}$$

for matrices of the same size. For a linear map A, $\|A\|_{\mathrm{op}}$ is the norm induced by the Frobenius norm.

The following proposition show the properties of the gradient of factorized objective.

Proposition 2.1 (Mu & Klabjan (2026)). *For $V = \left[B^\top; A\right]^\top$, we have*

$$\|V\|_F^2 = \|B\|_F^2 + \|A\|_F^2 \quad \textit{and} \quad \|BA\|_F \le \|B\|_F \, \|A\|_F \le \frac{1}{2}\|V\|_F^2 . \tag{2}$$

Moreover, the gradient of $J(\cdot)$ is

$$\nabla J(V) = \begin{bmatrix} \nabla f(BA)A^\top \\ \nabla f(BA)^\top B \end{bmatrix}. \tag{3}$$

We consider to find the approximate first-order stationarity of $J(\cdot)$, which is defined in the following.

Definition 2.1. For $\epsilon \ge 0$, a point V is an *ϵ-first-order stationary point* of $J(\cdot)$ if $\|\nabla J(V)\|_F \le \epsilon$. Equivalently,

$$\|\nabla J(V)\|_F^2 = \left\|\nabla f(BA)A^\top\right\|_F^2 + \left\|B^\top \nabla f(BA)\right\|_F^2 \le \epsilon^2.$$

For deterministic methods, we report the number of iterations needed to return V with $\|\nabla J(V)\|_F \le \epsilon$. For stochastic methods, we report stochastic oracle complexity for returning a random $\widetilde{V}$ with $\mathbb{E}\left\|\nabla J(\widetilde{V})\right\|_F \le \epsilon$. Fix an arbitrary initialization V_0. We denote $\Delta := J(V_0) - f_\star \ge 0$.

We introduce the following standard assumptions on the loss $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ and the stochastic oracle.

Assumption 1. We assume that $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ is continuously differentiable. There exists a gradient-Lipschitz parameter $\rho \ge 1$ such that, for every $W, W' \in \mathbb{R}^{m\times n}$,

$$\|\nabla f(W) - \nabla f(W')\|_F \le \rho \|W - W'\|_F .$$

Assumption 2. There exists a finite lower bound $f_\star \in \mathbb{R}$ such that $f(W) \ge f_\star$ for every $W \in \mathbb{R}^{m\times n}$.

In the stochastic setting, we consider $f(\cdot)$ can be reformulated by

$$f(W) = \mathbb{E}_\xi[F(W;\xi)]. \tag{4}$$

We impose the standard unbiasedness and finite-variance conditions on $\nabla F(W;\xi)$.

Assumption 3. For every fixed $W \in \mathbb{R}^{m\times n}$, we assume that the stochastic gradient satisfies

$$\mathbb{E}_\xi[\nabla F(W;\xi)] = \nabla f(W) \quad \text{and} \quad \mathbb{E}_\xi[\|\nabla F(W;\xi) - \nabla f(W)\|_F^2] \le \sigma^2 \tag{5}$$

for some finite $\sigma \ge 0$.

We also introduce the following stochastic gradient oracle:

$$G(V;\xi) := \begin{bmatrix} \nabla F(BA;\xi)A^\top \\ \nabla F(BA;\xi)^\top B \end{bmatrix}.$$

The following lemma establishes unbiasedness and bounds the variance of $G(V;\xi)$.

Lemma 2.2. *Under Assumption 3, we have*

$$\mathbb{E}_\xi[G(V;\xi)] = \nabla J(V) \quad \textit{and} \quad \mathbb{E}_\xi[\|G(V;\xi) - \nabla J(V)\|_F^2] \le \sigma^2 \|V\|_F^2 . \tag{6}$$

3 LoRA GRADIENT DESCENT AND ITS IMPROVED ITERATION COMPLEXITY

In this section, we sharpen the convergence analysis of the LoRA gradient descent (Algorithm 1).

Algorithm 1 LoRA gradient descent (Mu & Klabjan, 2026)

Require: $V_0 = \begin{bmatrix} B_0 \\ A_0^\top \end{bmatrix}$, horizon $T \geq 1$

1: **for** $t = 0, \ldots, T-1$ **do**
2: $\quad d_t \leftarrow \|V_t\|_F^2 + \|\nabla f(B_t A_t)\|_F$
3: $\quad \eta_t \leftarrow \min\{(4\sqrt{2}\,\rho d_t)^{-1}, 1\}$
4: $\quad V_{t+1} \leftarrow V_t - \eta_t \begin{bmatrix} \nabla f(B_t A_t) A_t^\top \\ \nabla f(B_t A_t)^\top B_t \end{bmatrix}$
5: **end for**
6: $\widehat{t} \in \arg\min_{0 \leq t < T} \|\nabla J(V_t)\|_F^2$
7: **return** $V_{\widehat{t}}$

Mu & Klabjan (2026) analyze Algorithm 1 through the equivalent stacked update

$$V_{t+1} = V_t - \eta_t \nabla J(V_t),$$

where $V_t := [B_t^\top; A_t]^\top$. Their descent estimate yields the weighted budget

$$\sum_{t=0}^{T-1} \eta_t \|\nabla J(V_t)\|_F^2 \leq 4\Delta,$$

so the rate is determined by the cumulative stepsize $\sum_{t<T} \eta_t$. Mu and Klabjan bound the cross term in the factor-norm recursion separately at every iteration using Young's inequality. This gives $\|V_t\|_F^2 = \mathcal{O}(t+1)$, and hence $\eta_t = \Omega((t+1)^{-1})$. The resulting harmonic sum leads to $\min_{t<T} \|\nabla J(V_t)\|_F^2 = \mathcal{O}(1/\log T)$.

We sharpen their analysis by summing the factor-norm recursion before applying Cauchy–Schwarz and reusing the weighted descent budget. We obtain the tighter trajectory bound $\|V_t\|_F^2 = \mathcal{O}(\sqrt{t+1})$. Moreover, descent and smoothness give $\|\nabla f(B_t A_t)\|_F \leq \sqrt{2\rho\Delta}$. It follows that the stepsize denominator satisfies $d_t = \mathcal{O}(\sqrt{t+1})$, and hence $\eta_t = \Omega((t+1)^{-1/2})$. Therefore, $\sum_{t<T} \eta_t = \Omega(\sqrt{T})$, and the same weighted budget gives $\min_{t<T} \|\nabla J(V_t)\|_F = \mathcal{O}(T^{-1/4})$. This sharper trajectory estimate yields the following improved iteration complexity for LoRA-GD.

Theorem 3.1. *Under Assumptions 1 and 2, for any $\epsilon \in (0, 1]$, it suffices to run Algorithm 1 with*

$$T = \mathcal{O}\left(\left\{ 1 + \Delta^2 \left[1 + \rho \|V_0\|_F^2 + \rho\Delta + \rho^{3/2}\sqrt{\Delta} \right]^2 \right\} \epsilon^{-4} \right).$$

The output then satisfies $\|\nabla J(V_{\widehat{t}})\|_F \leq \epsilon$.

4 LoRA NORMALIZED STOCHASTIC GRADIENT DESCENT WITH MOMENTUM

In this section, we establish convergence guarantees for stochastic LoRA. We first show that LoRA-SGD may fail to converge under only a finite-variance assumption. We then propose normalized stochastic gradient descent with momentum (NSGDM) and prove a non-asymptotic convergence guarantee under the same mild assumptions.

4.1 LoRA-SGD FAILS TO CONVERGE

We first show why Assumptions 1, 2, and 3 do not suffice for stochastic gradient descent, which updates according to

$$V_{t+1} = V_t - \eta_t \begin{bmatrix} \nabla F(B_t A_t; \xi_t) A_t^\top \\ \nabla F(B_t A_t; \xi_t)^\top B_t \end{bmatrix}, \tag{7}$$

where $\eta_t > 0$ may depend on all previous samples and iterates but is chosen before the oracle query at iteration t. The following proposition shows that even when the stepsizes are chosen adaptively, the iterates of (7) may diverge in expectation.

Proposition 4.1. *There are a scalar loss, stochastic oracle, and initialization satisfying Assumptions 1, 2, and 3 such that, for every stepsize rule where $\eta_t > 0$ may depend on all previous samples and iterates before drawing ξ_t, the iterates of* (7) *satisfy, for every $t \geq 1$,*

$$\mathbb{E}[J(V_t)] = +\infty \quad \textit{and} \quad \mathbb{E}\left\|\nabla J(V_t)\right\|_F = +\infty. \tag{8}$$

The difficulty also appears in the analysis based on the descent lemma established in Mu & Klabjan (2026), which we restate in Lemma B.1 in Appendix B for completeness. The descent lemma with $U = -\eta_t G(V_t;\xi_t)$ gives

$$J(V_{t+1}) - J(V_t) \leq -\eta_t \left\langle \nabla J(V_t), G(V_t;\xi_t)\right\rangle_F + \eta_t^2\left(\sqrt{2}\rho\left\|V_t\right\|_F^2 + \left\|\nabla f(B_tA_t)\right\|_F\right)\left\|G(V_t;\xi_t)\right\|_F^2$$
$$+ \sqrt{2}\rho\eta_t^3\left\|V_t\right\|_F\left\|G(V_t;\xi_t)\right\|_F^3 + \frac{\sqrt{2}\rho}{4}\eta_t^4\left\|G(V_t;\xi_t)\right\|_F^4. \tag{9}$$

Assumption 3 controls only the second moment of the oracle noise and therefore does not control the cubic and quartic terms in (9), which is regarded as a central challenge by Mu & Klabjan (2026).

4.2 Normalized SGD with Momentum Converges for Stochastic LoRA

The difficulty above originates from the random length $\eta_t\left\|G(V_t;\xi_t)\right\|_F$ of a plain LoRA-SGD step, which produces the cubic and quartic stochastic-gradient terms in (9). We first normalize the update so that every step has the prescribed length γ. This removes the need to control higher-order moments of the stochastic gradient. On the other hand, Lemma 2.2 gives

$$\mathbb{E}_{\xi_t}\left[\left\|G(V_t;\xi_t) - \nabla J(V_t)\right\|_F^2\right] \leq \sigma^2\left\|V_t\right\|_F^2,$$

which means that $G(V_t;\xi_t)$ may become increasingly noisy as $\left\|V_t\right\|_F$ grows. We therefore use momentum to average the stochastic gradients before normalizing the resulting direction to reduce the variance. This leads to the normalized stochastic gradient descent with momentum method (Cutkosky & Mehta, 2020) for the stochastic LoRA presented in Algorithm 2.

Algorithm 2 LoRA-NSGDM

Require: $V_0 = \begin{bmatrix} B_0 \\ A_0^\top \end{bmatrix}$, $T \geq 1$

1: $\alpha \leftarrow T^{-1/2}$, $\gamma \leftarrow T^{-7/8}$, and $M_{-1} \leftarrow 0$
2: **for** $t = 0, \ldots, T-1$ **do**
3: $\quad G(V_t;\xi_t) \leftarrow \begin{bmatrix} \nabla F(B_tA_t;\xi_t)A_t^\top \\ \nabla F(B_tA_t;\xi_t)^\top B_t \end{bmatrix}$
4: $\quad M_t \leftarrow (1-\alpha)M_{t-1} + \alpha G(V_t;\xi_t)$
5: $\quad V_{t+1} \leftarrow V_t - \gamma\frac{M_t}{\left\|M_t\right\|_F}$
6: **end for**
7: Draw I uniformly from $\{0, \ldots, T-1\}$, independently of the oracle samples
8: **return** V_I

We next give a non-asymptotic analysis of Algorithm 2. We introduce the deterministic radius and local smoothness scale

$$R_T = \left\|V_0\right\|_F + T\gamma \quad \text{and} \quad L_T = \left\|\nabla f(0)\right\|_F + \frac{3\rho R_T^2}{2}. \tag{10}$$

The following lemma shows that the normalization controls the trajectory of the iterates.

Lemma 4.2. *Suppose Assumptions 1 and 3 hold. Fix any horizon $T \geq 1$, and let $\{V_t\}_{t=0}^T$ and $\{M_t\}_{t=-1}^{T-1}$ be generated by Algorithm 2. With R_T and L_T defined in* (10)*, all iterates lie in the ball*

$$V_t \in \mathcal{B}_{R_T} := \{V : \left\|V\right\|_F \leq R_T\}, \qquad 0 \leq t \leq T.$$

Moreover, J is L_T-smooth on this ball:

$$\left\|\nabla J(V) - \nabla J(U)\right\|_F \leq L_T\left\|V - U\right\|_F \qquad \textit{for all } U, V \in \mathcal{B}_{R_T}. \tag{11}$$

For every $0 \leq t < T$, the factor-gradient oracle satisfies

$$\mathbb{E}_{\xi_t}[G(V_t;\xi_t)] = \nabla J(V_t) \quad \textit{and} \quad \mathbb{E}_{\xi_t}\left\|G(V_t;\xi_t) - \nabla J(V_t)\right\|_F^2 \leq \sigma^2 R_T^2. \tag{12}$$

We next quantify how momentum balances its initialization error, the drift of the true gradient, and the averaged oracle noise.

Lemma 4.3. *Suppose Assumptions 1 and 3 hold. Fix any horizon $T \geq 1$, and let $\{V_t\}_{t=0}^{T}$ and $\{M_t\}_{t=-1}^{T-1}$ be generated by Algorithm 2. Then*

$$\frac{1}{T}\sum_{t<T}\mathbb{E}\left\|M_t - \nabla J(V_t)\right\|_F \leq \frac{\left\|\nabla J(V_0)\right\|_F}{\alpha T} + \frac{L_T\gamma}{\alpha} + \sigma R_T\sqrt{\alpha}. \tag{13}$$

Then, we show one-step descent lemma of LoRA-NSGDM on the objective $J(\,\cdot\,)$.

Lemma 4.4. *Suppose Assumption 1 holds. Fix any horizon $T \geq 1$, and let $\{V_t\}_{t=0}^{T}$ and $\{M_t\}_{t=-1}^{T-1}$ be generated by Algorithm 2. Then, for every $0 \leq t < T$,*

$$J(V_{t+1}) \leq J(V_t) - \gamma\left\|\nabla J(V_t)\right\|_F + 2\gamma\left\|M_t - \nabla J(V_t)\right\|_F + \frac{L_T\gamma^2}{2}. \tag{14}$$

Combining the above lemmas gives the convergence rate and oracle complexity of Algorithm 2.

Theorem 4.5. *Under Assumptions 1, 2, and 3, for any $\epsilon \in (0,1]$, it suffices to run Algorithm 2 with*

$$T = \mathcal{O}\left(\left[1 + \Delta + (1+\left\|V_0\right\|_F)\big(\left\|\nabla f(0)\right\|_F + \sigma\big) + \rho\big(\left\|V_0\right\|_F^3 + (1+\left\|V_0\right\|_F)^2\big)\right]^8 \epsilon^{-8}\right). \tag{15}$$

The output then satisfies $\mathbb{E}\left\|\nabla J(V_I)\right\|_F \leq \epsilon$. Since each iteration uses one stochastic oracle evaluation, its stochastic oracle complexity is also given by (15)*.*

5 LoRA-STORM WITH IMPROVED STOCHASTIC ORACLE COMPLEXITY

In this section, we study the convergence of stochatsic LoRA under the mean-square smoothness conditrion, which is widely used in the variance reduction methods (Reddi et al., 2016; Fang et al., 2018; Cutkosky & Orabona, 2019; Li et al., 2021).

Assumption 4. There is a finite $\ell \geq 0$ such that, for every $W, W' \in \mathbb{R}^{m\times n}$,

$$\mathbb{E}_\xi\left\|\nabla F(W;\xi) - \nabla F(W';\xi)\right\|_F^2 \leq \ell^2\left\|W - W'\right\|_F^2.$$

The mean-square smoothness condition yields a corresponding local bound for the oracle $G(V;\xi)$.

Proposition 5.1. *Suppose Assumptions 3 and 4 hold. For any $U, V \in \mathbb{R}^{(m+n)\times r}$, we have*

$$\mathbb{E}_\xi\left\|G(V;\xi) - G(U;\xi)\right\|_F^2 \leq \left[3\ell^2R^4 + 4\left\|\nabla f(0)\right\|_F^2 + 2\sigma^2\right]\left\|V - U\right\|_F^2, \tag{16}$$

where $R = \max\{\left\|U\right\|_F, \left\|V\right\|_F\}$.

Algorithm 3 LoRA-STORM

Require: $V_0 = \begin{bmatrix} B_0 \\ A_0^\top \end{bmatrix}$, horizon $T \geq 1$

$a \leftarrow T^{-2/3}$ and $\eta \leftarrow T^{-5/6}$

$D_0 \leftarrow \begin{bmatrix} \nabla F(B_0A_0;\xi_0)A_0^\top \\ \nabla F(B_0A_0;\xi_0)^\top B_0 \end{bmatrix}$

$V_1 \leftarrow V_0 - \eta\frac{D_0}{\left\|D_0\right\|_F}$

for $t = 1, \ldots, T-1$ **do**

  $G(V_t;\xi_t) \leftarrow \begin{bmatrix} \nabla F(B_tA_t;\xi_t)A_t^\top \\ \nabla F(B_tA_t;\xi_t)^\top B_t \end{bmatrix}$

  $G(V_{t-1};\xi_t) \leftarrow \begin{bmatrix} \nabla F(B_{t-1}A_{t-1};\xi_t)A_{t-1}^\top \\ \nabla F(B_{t-1}A_{t-1};\xi_t)^\top B_{t-1} \end{bmatrix}$

  $D_t \leftarrow G(V_t;\xi_t) + (1-a)\big(D_{t-1} - G(V_{t-1};\xi_t)\big)$

  $V_{t+1} \leftarrow V_t - \eta\frac{D_t}{\left\|D_t\right\|_F}$

end for

Draw I uniformly from $\{0, \ldots, T-1\}$, independently of the oracle samples

return V_I

Proposition 5.1 suggests using variance reduction to improve the stochastic oracle complexity of LoRA-NSGDM. We apply the STORM method Cutkosky & Orabona (2019) for LoRA in Algorithm 3, where the update direction is equivalent to

$$D_t = (1-a)D_{t-1} + aG(V_t;\xi_t) + (1-a)(G(V_t;\xi_t) - G(V_{t-1};\xi_t)) \tag{17}$$

We compare the update rule (17) with the LoRA-NSGDM update in Algorithm 2. The main difference is that LoRA-STORM uses an additional correction term $(1-a)(G(V_t;\xi_t) - G(V_{t-1};\xi_t))$ (Cutkosky & Orabona, 2019). This difference estimates the change in the true gradient between two consecutive iterates, and Proposition 5.1 controls its second moment. Since the normalized step gives $\|V_t - V_{t-1}\|_F = \eta$, the correction remains controlled throughout the run. The following theorem shows that LoRA-STORM finds an ϵ-stationary point with $\mathcal{O}(\epsilon^{-6})$ stochastic oracle complexity.

Theorem 5.2. *Under Assumptions 2, 3, and 4, for any $\epsilon \in (0,1]$, it suffices to run Algorithm 3 with*

$$T = \mathcal{O}\Big(\big[1+\Delta+(1+\|V_0\|_F)\big(\|\nabla f(0)\|_F+\sigma\big)+\ell(1+\|V_0\|_F)^2\big]^6\,\epsilon^{-6}\Big). \tag{18}$$

The output then satisfies $\mathbb{E}\,\|\nabla J(V_I)\|_F \le \epsilon$. Algorithm 3 uses $1+2(T-1) = 2T-1$ stochastic oracle evaluations, so its stochastic oracle complexity is also given by (18).

6 Experiments

We conduct experiments to verify the practical implications of our theory. In particular, we compare the convergence of LoRA-NSGDM and LoRA-STORM with LoRA-GD using the adaptive stepsizes of Mu & Klabjan (2026). We consider three tasks. The first trains logistic regression on pretrained image representations on CIFAR-10. The second trains a LoRA-parameterized ResNet-18 on CIFAR-10. The third fine-tunes TinyLlama-1.1B on Alpaca. We evaluate LoRA-STORM only in the logistic-regression experiment. Although LoRA-STORM offers improved stochastic-oracle complexity in theory, each iteration requires two stochastic-oracle evaluations. This additional cost makes LoRA-STORM less attractive for the larger-scale ResNet-18 and TinyLlama experiments, where we compare LoRA-NSGDM with the LoRA-GD baselines.

Unless stated otherwise, we follow the experimental protocol of Mu & Klabjan (2026), which includes the datasets, architectures, LoRA ranks, factor initializations, batch sizes, and training horizons. The baselines are LoRA-GD with their η^{adapt}, η^{adapt2}, and η^{norm} stepsizes. Following Mu & Klabjan (2026), we omit η^{adapt} from the ResNet-18 and TinyLlama experiments because computing the required product-space gradient incurs substantial memory overhead for large models. Within each experiment, all methods use the same data split, initial predictor, and minibatch order. Appendix G gives the complete experimental settings and hyperparameters.

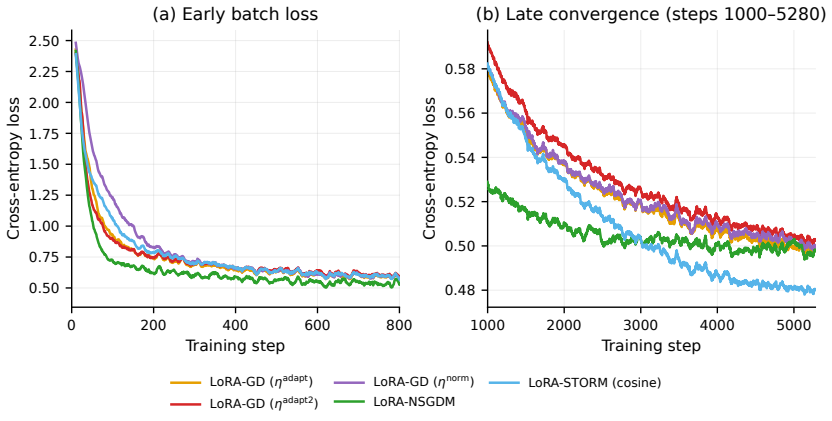


Figure 1: Training a rank-4 LoRA logistic-regression classifier on CIFAR-10 representations.

Logistic Regression. We first consider logistic regression on CIFAR-10 representations extracted by an ImageNet-pretrained Deng et al. (2009) ResNet-18 model He et al. (2016). Figure 1 presents the convergence comparison. Each CIFAR-10 image is passed once through the pretrained network. We use the resulting 512-dimensional penultimate-layer representation as the input to the linear classifier. These representations are computed before training and are shared by all methods. The classifier weight is $W_0 + BA$. The base matrix W_0 is fixed, and only the rank-4 factors A and B are trained. We initialize A with PyTorch's Kaiming-uniform rule, and set $B = 0$. Training uses cross-entropy, shuffled minibatches of size 512, no weight decay, and 88 steps per epoch. Panels (a) and (b) of Figure 1 show the first 60 epochs. In Figure 1a and 1b, LoRA-NSGDM exhibits faster convergence than the LoRA-GD algorithms. Over the final 2,000 updates shown in Figure 1(b), LoRA-STORM attains the lowest mean loss.

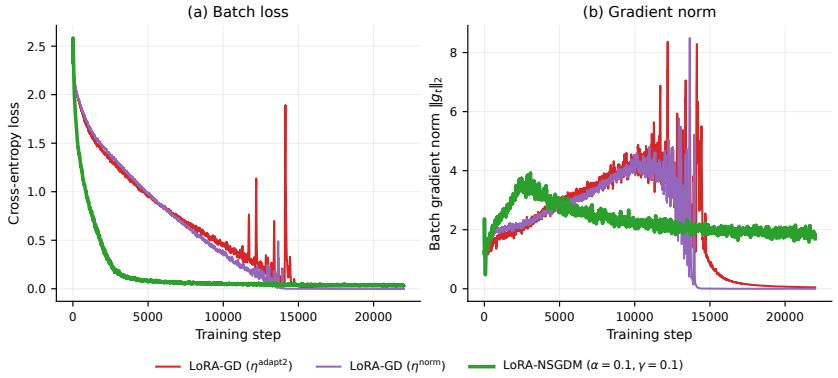


Figure 2: Training ResNet-18 on CIFAR-10 with rank-20 LoRA.

ResNet-18. We next train a ResNet-18 directly on CIFAR-10. Rank-20 LoRA factors are applied to the convolution layers. The output layer is trained without a rank constraint. Batch normalization layers are disabled following the settings in Mu & Klabjan (2026). We use minibatches of size 512, cross-entropy, and no weight decay. Each method is trained for 22,000 steps. Figure 2 shows the results over the first 250 epochs. LoRA-NSGDM converges significantly faster and more steadily than LoRA-GD algorithms.

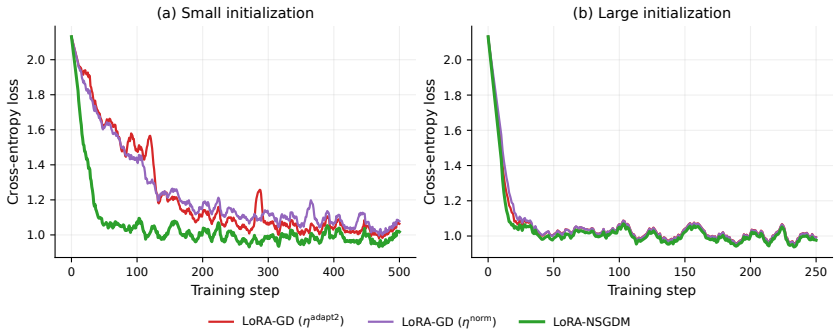


Figure 3: Fine-tuning TinyLlama-1.1B on Alpaca with rank-32 LoRA under (a) small and (b) large factor initialization.

TinyLlama-1.1B. Finally, we evaluate the methods on TinyLlama-1.1B instruction fine-tuning. Figure 3 compares small and large factor initialization. We fine-tune TinyLlama-1.1B-Chat-v1.0 (Zhang et al., 2024) on Alpaca (Taori et al., 2023). Each example contains an instruction, an optional input, and a response. Sequences are padded or truncated to 512 tokens, and padding tokens are excluded from the causal-language-model loss. We reserve 2,000 examples for validation. Each step uses one minibatch of 16 examples, with no gradient accumulation. Rank-32 LoRA factors are inserted into the query, key, value, and output projections of every attention block. The base-model

weights and biases remain fixed. We set $B = 0$ and sample $A \sim \mathcal{N}(0, \sigma^2)$. Small initialization uses $\sigma = 10^{-3}$ and 500 steps. Large initialization uses $\sigma = 1/r = 0.03125$ and 250 steps. Weight decay is zero.

With the small initialization shown in Figure 3a, LoRA-NSGDM maintains a faster convergence rate compared to the LoRA-GD algorithms. On the other hand, in Figure 3b, large initialization accelerates all three methods and reduces the separation. This behavior is consistent with our analysis: normalized momentum is most useful during the noisy initial phase induced by small factors. With large initialization, the trajectories become similar much earlier.

7 Conclusion

We established finite-time convergence guarantees for LoRA optimization. For deterministic LoRA-GD, we provide a refined analysis that improves the previously exponential sufficient complexity to $\mathcal{O}(\epsilon^{-4})$ iterations for finding ϵ-stationary point of $J(\cdot)$. In the stochastic setting, we showed that plain LoRA-SGD can fail under finite variance alone and introduced LoRA-NSGDM, which achieve $\mathcal{O}(\epsilon^{-8})$ stochastic oracle complexity for $\mathbb{E}\left\|\nabla J(V)\right\|_F \leq \epsilon$. When the stochastic oracle satisfies mean-square smoothness, LoRA-STORM improves the stochastic oracle complexity to $\mathcal{O}(\epsilon^{-6})$. Together, these results give polynomial deterministic and stochastic convergence guarantees for LoRA under the minimal oracle assumptions.

AI Use Statement

We used LLMs to generate an initial version of the proof for Section 5. The authors checked and revised this proof. We also used LLMs to summarize the related works of LoRA. All other parts of the manuscript were written by the authors and then polished by LLMs. We did not use generative AI to generate data, design experiments, implement methods, or interpret experimental results. The authors reviewed the AI-assisted proof and take full responsibility for the final manuscript.

Ethics Statement

This work studies optimization methods and uses publicly available datasets and pretrained models. It involves no human subjects, private or sensitive personal data, or user study. To the best of our knowledge, it raises no additional ethical concerns beyond the responsible use of the studied models and datasets.

Reproducibility Statement

The manuscript states the optimization assumptions, algorithms, oracle models, and convergence criteria, and the appendix contains the complete proofs. The experimental appendix gives the data splits, model configurations, preprocessing, hyperparameter grids, evaluation procedures, and hardware and software environments used for the reported results.

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *ACL-IJCNLP*, 2021.

Ashok Cutkosky and Harsh Mehta. Momentum improves normalized SGD. In *ICML*, 2020.

Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex SGD. In *NeurIPS*, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *NeurIPS*, 2023.

Shu Ding, Yang Peng, Hangan Zhou, Xinyu Lu, Shangwei Chen, Junhua Huang, Mingxuan Yuan, and Wei Wang. Towards understanding the dynamics of low-rank adaptation. In *ICML*, 2026.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *NeurIPS*, 2018.

Sajjad Ghiasvand, Mahnoosh Alizadeh, and Ramtin Pedarsani. Decentralized low-rank fine-tuning of large language models. In *REALM Workshop*, 2025.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low rank adaptation of large models. In *ICML*, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

Uijeong Jang, Jason D. Lee, and Ernest K. Ryu. LoRA training in the NTK regime has no spurious local minima. In *ICML*, 2024.

Zhanhong Jiang, Nastaran Saadati, Aditya Balu, Minh Pham, Joshua R. Waite, Nasla Saleem, Chinmay Hegde, and Soumik Sarkar. A unified convergence theory for large language model efficient fine-tuning. In *NeurIPS OPT Workshop*, 2024.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NeurIPS*, 2013.

Junsu Kim, Jaeyeon Kim, and Ernest K. Ryu. LoRA training provably converges to a low-rank global minimum or it fails loudly (but it probably won't fail). In *ICML*, 2025.

Zhize Li, Hongyan Bao, Xiangliang Zhang, and Peter Richtárik. PAGE: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In *ICML*, 2021.

Grigory Malinovsky, Umberto Michieli, Hasan Abed Al Kader Hammoud, Taha Ceritli, Hayder Elesedy, Mete Ozay, and Peter Richtárik. Randomized asymmetric chain of LoRA: The first meaningful theoretical framework for low-rank adaptation. *arXiv preprint arXiv:2410.08305*, 2024. doi: 10.48550/arXiv.2410.08305.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. PiSSA: Principal singular values and singular vectors adaptation of large language models. In *NeurIPS*, 2024.

Siqiao Mu and Diego Klabjan. On the convergence rate of LoRA gradient descent. In *ICML*, 2026.

Lam M. Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *ICML*, 2017.

Nhan H. Pham, Lam M. Nguyen, Dzung T. Phan, and Quoc Tran-Dinh. ProxSARAH: An efficient algorithmic framework for stochastic composite nonconvex optimization. *Journal of Machine Learning Research*, 21(110):1–48, 2020.

Sashank J. Reddi, Ahmed Hefny, Suvrit Sra, Barnabás Póczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *ICML*, 2016.

Igor Sokolov, Abdurakhmon Sadiev, Yury Demidovich, Fawaz S. Al-Qahtani, and Peter Richtárik. Bernoulli-LoRA: A theoretical framework for randomized low-rank adaptation. *arXiv preprint arXiv:2508.03820*, 2025. doi: 10.48550/arXiv.2508.03820.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving LoRA in privacy-preserving federated learning. In *ICLR*, 2024.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following LLaMA model, 2023.

Shaowen Wang, Linxi Yu, and Jian Li. LoRA-GA: Low-rank adaptation with gradient approximation. In *NeurIPS*, 2024.

Yangyang Xu and Yibo Xu. Momentum-based variance-reduced proximal stochastic gradient method for composite nonconvex stochastic optimization. *Journal of Optimization Theory and Applications*, 196(1):266–297, 2023. doi: 10.1007/s10957-022-02132-w.

Ziqing Xu, Hancheng Min, Lachlan Ewen MacDonald, Jinqi Luo, Salma Tarmoun, Enrique Mallada, and Rene Vidal. Understanding the learning dynamics of LoRA: A gradient flow perspective on low-rank adaptation in matrix factorization. In *AISTATS*, 2025.

Yuchen Zeng and Kangwook Lee. The expressive power of low-rank adaptation. In *ICLR*, 2024.

Fangzhao Zhang and Mert Pilanci. Riemannian preconditioned LoRA for fine-tuning foundation models. In *ICML*, 2024.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. TinyLlama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024. doi: 10.48550/arXiv.2401.02385.

Yuanhe Zhang, Fanghui Liu, and Yudong Chen. LoRA-one: One-step full gradient could suffice for fine-tuning large language models, provably and efficiently. In *ICML*, 2025.

A Related Work

Variants of LoRA. Adapting pretrained language models to downstream tasks can be costly when all model parameters are updated. Low-rank adaptation (LoRA) addresses this problem by freezing the pretrained weights and learning low-rank updates, substantially reducing the number of trainable parameters and the memory required for optimizer states (Hu et al., 2022). This approach is consistent with evidence that pretrained language models can be effectively fine-tuned in low-dimensional parameter subspaces (Aghajanyan et al., 2021). Subsequent variants of LoRA target memory efficiency through quantization (Dettmers et al., 2023) and optimization efficiency through factor-specific learning rates (Hayou et al., 2024). Initialization strategies that incorporate information from pretrained weights or full fine-tuning gradients can further improve the performance of LoRA (Meng et al., 2024; Wang et al., 2024). LoRA can also be extended to privacy-preserving federated learning and decentralized fine-tuning (Sun et al., 2024; Ghiasvand et al., 2025).

Convergence analysis of LoRA. Existing theory studies LoRA's optimization landscape and convergence under several sets of assumptions. In the neural tangent kernel (NTK) regime, Jang et al. (2024) establish the existence of low-rank optimal solutions and show that sufficiently large adapter ranks eliminate spurious local minima after a generic small perturbation of the objective. Beyond linearized models, Kim et al. (2025) characterize second-order stationary points under restricted strong convexity and smoothness when a low-rank global optimum exists, and show that spurious solutions must have sufficiently large rank and magnitude. For matrix factorization, Xu et al. (2025) analyze continuous-time gradient flow and characterize the dependence of the limiting error on the initialization scale and singular-subspace alignment. Initialization also plays a central role in LoRA-One: Zhang et al. (2025) establish linear convergence from spectral initialization for linear models and for preconditioned updates in a nonlinear model, under assumptions on the data distribution and the low-rank change between tasks. For stochastic LoRA, Jiang et al. (2024) obtain stationary-point guarantees under bounded factor singular values, bounded full gradients, and bounded stochastic-gradient variance in the factor variables. Another line of work modifies the factor updates. RAC-LoRA (Malinovsky et al., 2024) and Bernoulli-LoRA (Sokolov et al., 2025) sample and freeze one factor while optimizing the other, then accumulate the resulting low-rank update into the weight matrix. These frameworks establish stationarity guarantees for smooth nonconvex losses and linear convergence of their gradient-descent variants under the Polyak–Łojasiewicz condition. For simultaneous gradient updates of both factors, Mu & Klabjan (2026) assume only smoothness and lower boundedness of the original loss and prove an $\mathcal{O}(1/\log T)$ bound on $\min_{0\le t<T} \|\nabla J(V_t)\|_F^2$, with a stepsize adapted to the current factors and gradient. Their analysis handles the lack of global smoothness after factorization without assuming uniformly bounded factors. For a broader review of LoRA convergence theory, we refer to Mu & Klabjan (2026, Section 2).

Variance reduction for nonconvex optimization. Variance-reduction methods improve the oracle complexity of nonconvex stochastic optimization. SVRG (Johnson & Zhang, 2013) and SARAH (Nguyen et al., 2017) were introduced for smooth convex finite-sum optimization, with linear convergence guarantees under strong convexity. For smooth nonconvex finite-sum objectives, Reddi et al. (2016) establish non-asymptotic stationarity guarantees for SVRG. Recursive gradient estimators subsequently yield sharper guarantees, including SPIDER (Fang et al., 2018) and PAGE (Li et al., 2021). For lower-bounded online objectives with global mean-square smoothness and uniformly bounded stochastic-gradient variance, these two methods find a point whose expected gradient norm is at most ϵ using $\mathcal{O}(\epsilon^{-3})$ stochastic gradient evaluations. Their recursive updates estimate gradient changes by evaluating stochastic gradients at two successive iterates using the same sample. STORM incorporates such a correction into momentum and avoids periodic large-batch gradient computations (Cutkosky & Orabona, 2019). ProxSARAH and PStorm extend recursive variance reduction to composite nonconvex problems, with guarantees stated in terms of the proximal gradient mapping (Pham et al., 2020; Xu & Xu, 2023). Applying these stochastic guarantees requires smoothness and variance conditions in the variables being optimized. For LoRA, global smoothness of f does not imply global smoothness of J, and the factor-gradient variance bound grows with $\|V\|_F^2$ (Lemma 2.2). The mean-square smoothness bound for the factor oracle also depends on the factor norms (Proposition 5.1). Our analysis controls these quantities along the normalized factor trajectory to establish convergence of LoRA-STORM.

B Auxiliary Lemmas

Lemma B.1 (Mu & Klabjan, 2026, Lemma 3.3)**.** *Under Assumption 1, for every $V, U \in \mathbb{R}^{(m+n)\times r}$, we have*

$$J(V+U) \le J(V) + \langle \nabla J(V), U\rangle_F + \sqrt{2}\,\rho\,\|U\|_F^2\,\|V\|_F^2 + \sqrt{2}\,\rho\,\|U\|_F^3\,\|V\|_F + \frac{\sqrt{2}\,\rho}{4}\,\|U\|_F^4 + \|\nabla f(BA)\|_F\,\|U\|_F^2 .$$

Lemma B.2 (Mu & Klabjan, 2026, Lemma 3.4)**.** *Under Assumption 1, let $V_+ = V - \eta\nabla J(V)$, where the stepsize satisfies*

$$0 < \eta \le \min\left\{\frac{1}{4\sqrt{2}\,\rho(\|V\|_F^2 + \|\nabla f(BA)\|_F)}, 1\right\}.$$

Then we have

$$J(V_+) \le J(V) - \frac{\eta}{4}\,\|\nabla J(V)\|_F^2 .$$

Lemma B.3 (Mu & Klabjan, 2026, Eq. (23))**.** *If Assumptions 1 and 2 hold, then we have*

$$\|\nabla f(W)\|_F^2 \le 2\rho(f(W) - f_\star).$$

C Proof of Section 2

C.1 Proof of Lemma 2.2

Proof. Fix $V = \begin{bmatrix} B \\ A^\top \end{bmatrix}$. By the definition of $G(V;\xi)$ and Assumption 3, we have

$$\mathbb{E}_\xi[G(V;\xi)] = \begin{bmatrix} \mathbb{E}_\xi[\nabla F(BA;\xi)]A^\top \\ \mathbb{E}_\xi[\nabla F(BA;\xi)]^\top B \end{bmatrix} = \begin{bmatrix} \nabla f(BA)A^\top \\ \nabla f(BA)^\top B \end{bmatrix} = \nabla J(V).$$

Moreover, submultiplicativity of the Frobenius norm gives

$$\begin{aligned} &\|G(V;\xi) - \nabla J(V)\|_F^2 \\ &= \left\|(\nabla F(BA;\xi) - \nabla f(BA))A^\top\right\|_F^2 + \left\|(\nabla F(BA;\xi) - \nabla f(BA))^\top B\right\|_F^2 \\ &\le \|\nabla F(BA;\xi) - \nabla f(BA)\|_F^2\,\left(\|A\|_F^2 + \|B\|_F^2\right). \end{aligned}$$

Taking expectation and applying Assumption 3 at $W = BA$, we obtain

$$\begin{aligned} &\mathbb{E}_\xi\,\|G(V;\xi) - \nabla J(V)\|_F^2 \\ &\le \left(\|A\|_F^2 + \|B\|_F^2\right)\mathbb{E}_\xi\,\|\nabla F(BA;\xi) - \nabla f(BA)\|_F^2 \\ &\le \sigma^2\left(\|A\|_F^2 + \|B\|_F^2\right) = \sigma^2\,\|V\|_F^2 . \end{aligned}$$

□

D Proof of Section 3

D.1 Proof of Theorem 3.1

Proof. Set $g_t = \|\nabla J(V_t)\|_F$ and $\Delta = J(V_0) - f_\star$. The stepsize in Algorithm 1 satisfies the hypothesis of Lemma B.2. Applying the lemma at each iteration and summing gives

$$\sum_{t=0}^{T-1} \eta_t g_t^2 \le 4\Delta. \tag{19}$$

Thus $J(V_t) \le J(V_0)$, and Lemma B.3, applied at B_tA_t, yields

$$\|\nabla f(B_tA_t)\|_F \le \sqrt{2\rho\Delta}. \tag{20}$$

The definition of η_t gives

$$\eta_t \|V_t\|_F^2 \le \frac{1}{4\sqrt{2}\,\rho} \quad \text{and} \quad \eta_t \le 1. \tag{21}$$

The update and Cauchy–Schwarz yield

$$\begin{aligned}\|V_{t+1}\|_F^2 &\le \|V_t\|_F^2 + 2\eta_t \|V_t\|_F g_t + \eta_t^2 g_t^2 \\ &\le \|V_t\|_F^2 + \frac{2\sqrt{\eta_t} g_t}{\sqrt{4\sqrt{2}\,\rho}} + \eta_t g_t^2.\end{aligned}$$

Summing over $s = 0, \dots, t-1$, applying Cauchy–Schwarz, and using (19) yield

$$\begin{aligned}\|V_t\|_F^2 &\le \|V_0\|_F^2 + \frac{2}{\sqrt{4\sqrt{2}\,\rho}} \sum_{s=0}^{t-1} \sqrt{\eta_s} g_s + \sum_{s=0}^{t-1} \eta_s g_s^2 \\ &\le \|V_0\|_F^2 + 2\sqrt{\frac{t}{4\sqrt{2}\,\rho}} \left(\sum_{s=0}^{t-1} \eta_s g_s^2 \right)^{1/2} + \sum_{s=0}^{t-1} \eta_s g_s^2 \\ &\le \|V_0\|_F^2 + 4\sqrt{\frac{\Delta}{4\sqrt{2}\,\rho}}\,\sqrt{t} + 4\Delta.\end{aligned} \tag{22}$$

Define

$$C_0 = \|V_0\|_F^2 + 4\Delta + \sqrt{2\rho\Delta} \quad \text{and} \quad C_1 = 4\sqrt{\frac{\Delta}{4\sqrt{2}\,\rho}}.$$

Equations (20) and (22) give

$$d_t = \|V_t\|_F^2 + \|\nabla f(B_t A_t)\|_F \le C_0 + C_1\sqrt{t}.$$

Writing the stepsize in Algorithm 1 as the reciprocal of a maximum and substituting this bound, we obtain

$$\begin{aligned}\eta_t &= \frac{1}{\max\{1, 4\sqrt{2}\,\rho d_t\}} \\ &\ge \frac{1}{\max\{1, 4\sqrt{2}\,\rho(C_0 + C_1\sqrt{t})\}} \\ &\ge \frac{1}{1 + 4\sqrt{2}\,\rho(C_0 + C_1\sqrt{t})}.\end{aligned} \tag{23}$$

Since $\sqrt{t} \le \sqrt{T}$ for $0 \le t < T$, we have

$$\sum_{t=0}^{T-1} \eta_t \ge \frac{T}{1 + 4\sqrt{2}\,\rho(C_0 + C_1\sqrt{T})}. \tag{24}$$

Equation (19) implies

$$\min_{0 \le t < T} g_t^2 \le \frac{4\Delta}{\sum_{t=0}^{T-1} \eta_t}.$$

Substituting (24) gives

$$\min_{0 \le t < T} g_t^2 \le \frac{4\Delta \left[1 + 4\sqrt{2}\,\rho(C_0 + C_1\sqrt{T})\right]}{T}. \tag{25}$$

Set

$$C := \max\left\{1,\ 16\Delta^2 \left[1 + 4\sqrt{2}\,\rho\left(\|V_0\|_F^2 + 4\Delta + \sqrt{2\rho\Delta} + 4\sqrt{\frac{\Delta}{4\sqrt{2}\,\rho}} \right)\right]^2\right\}. \tag{26}$$

For $T \ge 1$, the right side of (25) is at most

$$4\Delta \left[1 + 4\sqrt{2}\,\rho(C_0 + C_1)\right] T^{-1/2} \le \sqrt{C}\, T^{-1/2}.$$

Thus, choosing $T = \lceil C\epsilon^{-4} \rceil$ gives $\|\nabla J(V_{\widehat{t}})\|_F \le \epsilon$. □

E Proof of Section 4

E.1 Proof of Proposition 4.1

Proof. Let $m = n = r = 1$, write the factors as $b, a \in \mathbb{R}$, and take

$$F(w;\xi) = \frac{1}{2}w^2 + \xi w, \qquad f(w) = \mathbb{E}_\xi[F(w;\xi)] = \frac{1}{2}w^2, \qquad \text{and} \qquad \nabla F(w;\xi) = w + \xi.$$

Let ξ have the continuous symmetric density

$$p(z) = \frac{3}{2(1+|z|)^4}, \qquad z \in \mathbb{R}.$$

Direct integration gives

$$\mathbb{E}[\xi] = 0, \qquad \mathbb{E}[\xi^2] = 1, \qquad \text{and} \qquad \mathbb{E}[|\xi|^q] = +\infty \quad \text{for every } q \geq 3.$$

Thus Assumption 1 holds with $\rho = 1$, Assumption 2 holds with $f_\star = 0$, and Assumption 3 holds with $\sigma^2 = 1$.

Initialize $b_0 = a_0 = 1$. Equality of the two factors is preserved by (7); writing $b_t = a_t = x_t$ gives

$$x_{t+1} = x_t\big[1 - \eta_t(x_t^2 + \xi_t)\big]. \tag{27}$$

For fixed $x_t \neq 0$ and $\eta_t > 0$, continuity of the density gives

$$\Pr_{\xi_t}(x_{t+1} = 0) = \Pr_{\xi_t}(\xi_t = \eta_t^{-1} - x_t^2) = 0.$$

Thus $x_t \neq 0$ almost surely for every t, by induction from $x_0 = 1$.

Under the same conditioning, x_t and η_t are fixed, and (27) becomes

$$x_{t+1} = c_t + d_t\xi_t, \qquad c_t := x_t(1 - \eta_t x_t^2), \qquad \text{and} \qquad d_t := -\eta_t x_t.$$

The coefficient d_t is nonzero almost surely. Set $R_t = 2|c_t|/|d_t| < \infty$. Whenever $|\xi_t| \geq R_t$, the reverse triangle inequality gives

$$|x_{t+1}| = |c_t + d_t\xi_t| \geq |d_t||\xi_t| - |c_t| \geq \frac{|d_t|}{2}|\xi_t|.$$

Therefore, conditioning on the past gives

$$\mathbb{E}_{\xi_t}[|x_{t+1}|^3] \geq \frac{|d_t|^3}{8}\mathbb{E}_{\xi_t}\big[|\xi_t|^3\mathbf{1}\{|\xi_t| \geq R_t\}\big] = +\infty.$$

The last equality follows because removing any bounded interval from a random variable with $\mathbb{E}|\xi_t|^3 = +\infty$ leaves an infinite third-moment tail. Since this conditional expectation is infinite almost surely, averaging over the past gives $\mathbb{E}|x_{t+1}|^3 = +\infty$. The same argument with fourth powers gives $\mathbb{E}|x_{t+1}|^4 = +\infty$. Finally, we have

$$J(x_t, x_t) = \frac{1}{2}x_t^4, \qquad \nabla J(x_t, x_t) = \begin{bmatrix} x_t^3 \\ x_t^3 \end{bmatrix}, \qquad \text{and} \qquad \|\nabla J(x_t, x_t)\|_F = \sqrt{2}\,|x_t|^3.$$

Applying these identities at $t+1$ proves (8) for every $t \geq 1$. □

E.2 Proof of Lemma 4.2

Proof. Every update in Algorithm 2 has length at most γ. The triangle inequality therefore gives

$$\|V_t\|_F \leq \|V_0\|_F + t\gamma \leq R_T.$$

For every $0 \leq t < T$, each point on the line segment from V_t to V_{t+1} has the form $(1-\lambda)V_t + \lambda V_{t+1}$, where $\lambda \in [0,1]$. By the triangle inequality, we have

$$\|(1-\lambda)V_t + \lambda V_{t+1}\|_F \leq (1-\lambda)\,\|V_t\|_F + \lambda\,\|V_{t+1}\|_F \leq R_T.$$

Thus the entire line segment lies in $\mathcal{B}_{R_T}$.

Let

$$V = \begin{bmatrix} B \\ A^\top \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} \widetilde{B} \\ \widetilde{A}^\top \end{bmatrix}$$

be in $\mathcal{B}_{R_T}$. We have

$$BA - \widetilde{B}\widetilde{A} = \frac{1}{2}\big((B - \widetilde{B})(A + \widetilde{A}) + (B + \widetilde{B})(A - \widetilde{A})\big).$$

Submultiplicativity and Cauchy–Schwarz therefore give

$$\left\|BA - \widetilde{B}\widetilde{A}\right\|_F \le \frac{1}{2}\|V - U\|_F \|V + U\|_F \le R_T \|V - U\|_F . \tag{28}$$

Proposition 2.1 also gives $\|BA\|_F \le \|V\|_F^2/2$. Hence, by Assumption 1, we have

$$\sup_{V \in \mathcal{B}_{R_T}} \|\nabla f(BA)\|_F \le \|\nabla f(0)\|_F + \frac{\rho R_T^2}{2}.$$

Applying the triangle inequality and submultiplicativity to the factor-gradient formula (3), and then using (28) and the preceding bound, we obtain

$$\begin{aligned}
&\|\nabla J(V) - \nabla J(U)\|_F \\
&\le \left\|\nabla f(BA) - \nabla f(\widetilde{B}\widetilde{A})\right\|_F \|V\|_F + \left\|\nabla f(\widetilde{B}\widetilde{A})\right\|_F \|V - U\|_F \\
&\le \left(\|\nabla f(0)\|_F + \frac{3\rho R_T^2}{2}\right)\|V - U\|_F ,
\end{aligned}$$

which proves (11).

Finally, Lemma 2.2 and $\|V_t\|_F \le R_T$ give

$$\mathbb{E}_{\xi_t}[G(V_t;\xi_t) - \nabla J(V_t)] = 0 \quad \text{and} \quad \mathbb{E}_{\xi_t}\|G(V_t;\xi_t) - \nabla J(V_t)\|_F^2 \le \sigma^2 \|V_t\|_F^2 \le \sigma^2 R_T^2,$$

which proves (12). □

E.3 Proof of Lemma 4.3

Proof. Let

$$G_t := \nabla J(V_t), \qquad Z_t := G(V_t;\xi_t) - G_t, \qquad \text{and} \qquad E_t := M_t - G_t,$$

and let $q := 1 - \alpha$. Lemma 4.2 gives

$$\mathbb{E}_{\xi_t}[Z_t] = 0 \quad \text{and} \quad \mathbb{E}_{\xi_t}\|Z_t\|_F^2 \le \sigma^2 R_T^2.$$

For $t \ge 1$, we have

$$E_t = qE_{t-1} + q(G_{t-1} - G_t) + \alpha Z_t \quad \text{and} \quad E_0 = -qG_0 + \alpha Z_0.$$

Unrolling gives

$$E_t = -q^{t+1}G_0 + q\sum_{s=1}^{t} q^{t-s}(G_{s-1} - G_s) + \alpha\sum_{s=0}^{t} q^{t-s}Z_s. \tag{29}$$

Local smoothness and the step-length bound imply $\|G_s - G_{s-1}\|_F \le L_T\gamma$. Hence the drift term in (29) satisfies

$$\left\|q\sum_{s=1}^{t} q^{t-s}(G_{s-1} - G_s)\right\|_F \le qL_T\gamma\sum_{k=0}^{t-1} q^k \le \frac{L_T\gamma}{\alpha},$$

since $1 - q = \alpha$.

For $r < s$, conditional unbiasedness gives

$$\mathbb{E}\,\langle Z_r, Z_s\rangle_F = \mathbb{E}_{\xi_0,\dots,\xi_{s-1}}\langle Z_r, \mathbb{E}_{\xi_s}[Z_s]\rangle_F = 0.$$

Consequently, we have

$$\begin{aligned}
&\mathbb{E}\left\|\alpha\sum_{s=0}^{t} q^{t-s} Z_s\right\|_F \le \left(\mathbb{E}\left\|\alpha\sum_{s=0}^{t} q^{t-s} Z_s\right\|_F^2\right)^{1/2} \\
&= \alpha\left(\sum_{s=0}^{t} q^{2(t-s)}\mathbb{E}\left\|Z_s\right\|_F^2\right)^{1/2} \\
&\le \alpha\sigma R_T\left(\sum_{k=0}^{t} q^{2k}\right)^{1/2} \\
&\le \sigma R_T\sqrt{\frac{\alpha}{2-\alpha}} \le \sigma R_T\sqrt{\alpha}.
\end{aligned}$$

Combining the preceding two estimates with (29) and the triangle inequality gives, for every $0 \le t < T$,

$$\mathbb{E}\left\|E_t\right\|_F \le q^{t+1}\left\|G_0\right\|_F + \frac{L_T\gamma}{\alpha} + \sigma R_T\sqrt{\alpha}.$$

Since $1 - q = \alpha$, the initialization weights obey

$$\sum_{t=0}^{T-1} q^{t+1} = \frac{q(1-q^T)}{1-q} \le \frac{1}{\alpha}.$$

Therefore, averaging the pointwise bound over $t = 0, \dots, T-1$ yields

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|E_t\right\|_F \le \frac{\left\|G_0\right\|_F}{T}\sum_{t=0}^{T-1} q^{t+1} + \frac{L_T\gamma}{\alpha} + \sigma R_T\sqrt{\alpha} \\
&\le \frac{\left\|G_0\right\|_F}{\alpha T} + \frac{L_T\gamma}{\alpha} + \sigma R_T\sqrt{\alpha}.
\end{aligned}$$

Substituting $E_t = M_t - \nabla J(V_t)$ and $G_0 = \nabla J(V_0)$ gives (13). □

E.4 Proof of Lemma 4.4

Proof. Let

$$G_t := \nabla J(V_t) \quad \text{and} \quad E_t := M_t - G_t.$$

If $M_t \neq 0$, then we have

$$\begin{aligned}
&\langle G_t, M_t/\left\|M_t\right\|_F\rangle_F = \langle M_t - E_t, M_t/\left\|M_t\right\|_F\rangle_F \\
&= \left\|M_t\right\|_F - \langle E_t, M_t/\left\|M_t\right\|_F\rangle_F \\
&\ge \left\|M_t\right\|_F - \left\|E_t\right\|_F \\
&\ge \left\|G_t\right\|_F - 2\left\|E_t\right\|_F.
\end{aligned} \tag{30}$$

The first inequality follows from Cauchy–Schwarz, and the second follows from $\left\|M_t\right\|_F \ge \left\|G_t\right\|_F - \left\|E_t\right\|_F$.

Every update segment is contained in $\mathcal{B}_{R_T}$. The local descent lemma, the update in Algorithm 2, and (30) therefore give, whenever $M_t \neq 0$,

$$\begin{aligned}
&J(V_{t+1}) \le J(V_t) + \langle G_t, V_{t+1} - V_t\rangle_F + \frac{L_T}{2}\left\|V_{t+1} - V_t\right\|_F^2 \\
&= J(V_t) - \gamma\langle G_t, M_t/\left\|M_t\right\|_F\rangle_F + \frac{L_T\gamma^2}{2} \\
&\le J(V_t) - \gamma\left\|G_t\right\|_F + 2\gamma\left\|E_t\right\|_F + \frac{L_T\gamma^2}{2}.
\end{aligned}$$

If $M_t = 0$, then $E_t = -G_t$, and the zero update satisfies the same final inequality. Substituting the definitions of G_t and E_t gives (14). □

E.5 Proof of Theorem 4.5

Proof. Let

$$G_t := \nabla J(V_t).$$

Summing (14) over $t = 0, \dots, T-1$, taking expectation, using $J(V_T) \geq f_\star$, and applying Lemma 4.3 yield

$$\frac{1}{T}\sum_{t<T} \mathbb{E}\left\|G_t\right\|_F \leq \frac{\Delta}{\gamma T} + \frac{2\left\|G_0\right\|_F}{\alpha T} + \frac{2L_T\gamma}{\alpha} + 2\sigma R_T\sqrt{\alpha} + \frac{L_T\gamma}{2}. \tag{31}$$

Let

$$\overline{R} = \left\|V_0\right\|_F + 1, \qquad \overline{L} = \left\|\nabla f(0)\right\|_F + \frac{3\rho\overline{R}^2}{2},$$
$$\text{and} \quad G_{\text{init}} = \left\|V_0\right\|_F\left(\left\|\nabla f(0)\right\|_F + \frac{\rho}{2}\left\|V_0\right\|_F^2\right).$$

Proposition 2.1 and the ρ-smoothness of f give $\left\|G_0\right\|_F \leq G_{\text{init}}$. With $\gamma = T^{-7/8}$, we have, for $T \geq 1$,

$$R_T \leq \overline{R}T^{1/8}, \qquad L_T \leq \overline{L}T^{1/4}, \qquad \text{and} \qquad \sigma R_T \leq \sigma\overline{R}T^{1/8}.$$

Substituting these inequalities and $\alpha = T^{-1/2}$ into (31) yields

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left\|G_t\right\|_F \leq \left(\Delta + 2G_{\text{init}} + 2\sigma\overline{R} + \frac{5}{2}\overline{L}\right)T^{-1/8}. \tag{32}$$

For an independent uniform I, the left side is $\mathbb{E}\left\|\nabla J(V_I)\right\|_F$. We take $C_{\text{mom}} = \Delta + 2G_{\text{init}} + 2\sigma\overline{R} + 5\overline{L}/2$. In particular, we have

$$C_{\text{mom}} = \mathcal{O}\Big(1 + \Delta + (1 + \left\|V_0\right\|_F)\big(\left\|\nabla f(0)\right\|_F + \sigma\big) + \rho\big(\left\|V_0\right\|_F^3 + (1 + \left\|V_0\right\|_F)^2\big)\Big).$$

The choice of T in (15) therefore makes this expectation at most ϵ. Since Algorithm 2 makes one stochastic oracle evaluation per iteration, the same bound gives its stochastic oracle complexity. □

F Proof of Section 5

F.1 Proof of Proposition 5.1

Proof. First, unbiasedness and Jensen's inequality imply that, for any $W, W' \in \mathbb{R}^{m\times n}$, we have

$$\left\|\nabla f(W) - \nabla f(W')\right\|_F^2 = \left\|\mathbb{E}_\xi[\nabla F(W;\xi) - \nabla F(W';\xi)]\right\|_F^2$$
$$\leq \mathbb{E}_\xi\left\|\nabla F(W;\xi) - \nabla F(W';\xi)\right\|_F^2 \leq \ell^2\left\|W - W'\right\|_F^2,$$

where the last inequality follows from Assumption 4. Thus ∇f is ℓ-Lipschitz continuous.

Let

$$V = \begin{bmatrix} B \\ A^\top \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} \widetilde{B} \\ \widetilde{A}^\top \end{bmatrix},$$

and set $R = \max\{\left\|U\right\|_F, \left\|V\right\|_F\}$. The argument for (28), with radius R, gives

$$\left\|BA - \widetilde{B}\widetilde{A}\right\|_F \leq \frac{1}{2}\left\|V - U\right\|_F\left\|V + U\right\|_F \leq R\left\|V - U\right\|_F.$$

By the definition of G, submultiplicativity, and $\left\|X+Y\right\|_F^2 \leq 2\left\|X\right\|_F^2 + 2\left\|Y\right\|_F^2$, we have

$$\mathbb{E}_\xi\left\|G(V;\xi) - G(U;\xi)\right\|_F^2$$
$$\leq 2\left\|V\right\|_F^2\,\mathbb{E}_\xi\left\|\nabla F(BA;\xi) - \nabla F(\widetilde{B}\widetilde{A};\xi)\right\|_F^2 + 2\left\|V-U\right\|_F^2\,\mathbb{E}_\xi\left\|\nabla F(\widetilde{B}\widetilde{A};\xi)\right\|_F^2.$$

Assumption 4 and the product-difference bound imply

$$\mathbb{E}_\xi \left\| \nabla F(BA;\xi) - \nabla F(\widetilde{B}\widetilde{A};\xi) \right\|_F^2 \le \ell^2 \left\| BA - \widetilde{B}\widetilde{A} \right\|_F^2 \le \ell^2 R^2 \left\| V - U \right\|_F^2 .$$

For the remaining second moment, Assumption 3 and the ℓ-Lipschitz continuity of ∇f give

$$\begin{aligned} &\mathbb{E}_\xi \left\| \nabla F(\widetilde{B}\widetilde{A};\xi) \right\|_F^2 \\ &\le \left\| \nabla f(\widetilde{B}\widetilde{A}) \right\|_F^2 + \sigma^2 \\ &\le \left(\left\| \nabla f(0) \right\|_F + \ell \left\| \widetilde{B}\widetilde{A} \right\|_F \right)^2 + \sigma^2 \\ &\le \left(\left\| \nabla f(0) \right\|_F + \frac{\ell R^2}{2} \right)^2 + \sigma^2, \end{aligned}$$

where the last inequality uses $\left\| \widetilde{B}\widetilde{A} \right\|_F \le \left\| U \right\|_F^2 / 2$. Combining these estimates, we obtain

$$\mathbb{E}_\xi \left\| G(V;\xi) - G(U;\xi) \right\|_F^2 \le 2 \left(\ell^2 R^4 + \left(\left\| \nabla f(0) \right\|_F + \frac{\ell R^2}{2} \right)^2 + \sigma^2 \right) \left\| V - U \right\|_F^2 . \tag{33}$$

Applying $2(a+b)^2 \le 4a^2 + 4b^2$ to (33) gives

$$\mathbb{E}_\xi \left\| G(V;\xi) - G(U;\xi) \right\|_F^2 \le \left(3\ell^2 R^4 + 4 \left\| \nabla f(0) \right\|_F^2 + 2\sigma^2 \right) \left\| V - U \right\|_F^2 ,$$

which is (16). □

F.2 Auxiliary tracking lemma

Lemma F.1. *Suppose Assumptions 3 and 4 hold, and let the iterates be generated by Algorithm 3. Set*

$$R_T = \left\| V_0 \right\|_F + T\eta, \qquad S_T = \sigma R_T, \qquad S_0 = \sigma \left\| V_0 \right\|_F ,$$

and

$$\Lambda_T^2 = 2\ell^2 R_T^4 + 2 \left[\left(\left\| \nabla f(0) \right\|_F + \frac{\ell R_T^2}{2} \right)^2 + \sigma^2 \right].$$

Then the estimator satisfies

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \left\| D_t - \nabla J(V_t) \right\|_F \le \frac{S_0}{aT} + \sqrt{2a}\, S_T + \frac{\Lambda_T \eta}{\sqrt{a}}. \tag{34}$$

Proof. Set

$$G_t := \nabla J(V_t) \quad \text{and} \quad E_t := D_t - G_t.$$

By the initialization in Algorithm 3, we have

$$\mathbb{E} \left\| E_0 \right\|_F^2 = \mathbb{E} \left\| G(V_0;\xi_0) - G_0 \right\|_F^2 \le \sigma^2 \left\| V_0 \right\|_F^2 = S_0^2.$$

For $t \ge 1$, set

$$\Xi_t = G(V_t;\xi_t) - G_t \quad \text{and} \quad \Xi_{t-1}^{(t)} = G(V_{t-1};\xi_t) - G_{t-1},$$

and set $\Theta_t = \Xi_t - \Xi_{t-1}^{(t)}$ and $q = 1 - a$. The estimator update in Algorithm 3 gives

$$D_t = G(V_t;\xi_t) + q\big(D_{t-1} - G(V_{t-1};\xi_t) \big).$$

Subtracting G_t and using $a = 1 - q$ therefore yields

$$E_t = qE_{t-1} + a\Xi_t + q\Theta_t. \tag{35}$$

Let

$$X_t := G(V_t;\xi_t) - G(V_{t-1};\xi_t).$$

Lemma 2.2 gives

$$\mathbb{E}_{\xi_t}[\Xi_t] = 0 \quad \text{and} \quad \mathbb{E}_{\xi_t} \|\Xi_t\|_F^2 \le S_T^2.$$

Moreover, we have

$$\mathbb{E}_{\xi_t}[X_t] = G_t - G_{t-1} \quad \text{and} \quad \Theta_t = X_t - \mathbb{E}_{\xi_t}[X_t].$$

Hence, by (33) and $\|V_t - V_{t-1}\|_F \le \eta$, we have $\mathbb{E}_{\xi_t}[\Theta_t] = 0$ and

$$\begin{aligned}\mathbb{E}_{\xi_t} \|\Theta_t\|_F^2 &= \mathbb{E}_{\xi_t} \|X_t - \mathbb{E}_{\xi_t}[X_t]\|_F^2 \\ &\le \mathbb{E}_{\xi_t} \|X_t\|_F^2 \le \Lambda_T^2 \|V_t - V_{t-1}\|_F^2 \le \Lambda_T^2 \eta^2.\end{aligned}$$

The cross term in (35) satisfies

$$\mathbb{E}_{\xi_t} \langle E_{t-1}, a\Xi_t + q\Theta_t \rangle_F = \langle E_{t-1}, a\mathbb{E}_{\xi_t}[\Xi_t] + q\mathbb{E}_{\xi_t}[\Theta_t] \rangle_F = 0.$$

Taking expectation and using $\|X+Y\|_F^2 \le 2\|X\|_F^2 + 2\|Y\|_F^2$, we obtain

$$\begin{aligned}\mathbb{E}\|E_t\|_F^2 &= q^2 \mathbb{E}\|E_{t-1}\|_F^2 + \mathbb{E}\|a\Xi_t + q\Theta_t\|_F^2 \\ &\le q^2 \mathbb{E}\|E_{t-1}\|_F^2 + 2a^2 \mathbb{E}\|\Xi_t\|_F^2 + 2q^2 \mathbb{E}\|\Theta_t\|_F^2 \\ &\le q^2 \mathbb{E}\|E_{t-1}\|_F^2 + 2a^2 S_T^2 + 2q^2 \Lambda_T^2 \eta^2.\end{aligned}$$

Since $1 - q^2 = a(2-a)$ and $2q^2 \le 2 - a$ for $a \in (0, 1]$, unrolling this recurrence gives

$$\begin{aligned}\mathbb{E}\|E_t\|_F^2 &\le q^{2t} S_0^2 + \left(2a^2 S_T^2 + 2q^2 \Lambda_T^2 \eta^2\right) \sum_{k=0}^{t-1} q^{2k} \\ &\le q^{2t} S_0^2 + 2a S_T^2 + \frac{\Lambda_T^2 \eta^2}{a}.\end{aligned}$$

Jensen's inequality and $\sqrt{x+y+z} \le \sqrt{x} + \sqrt{y} + \sqrt{z}$ give

$$\mathbb{E}\|E_t\|_F \le q^t S_0 + \sqrt{2a}\, S_T + \frac{\Lambda_T \eta}{\sqrt{a}}.$$

Averaging over $t = 0, \dots, T-1$ gives

$$\begin{aligned}\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\|D_t - \nabla J(V_t)\|_F &= \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\|E_t\|_F \\ &\le \frac{S_0}{T} \sum_{t=0}^{T-1} q^t + \sqrt{2a}\, S_T + \frac{\Lambda_T \eta}{\sqrt{a}} \\ &\le \frac{S_0}{aT} + \sqrt{2a}\, S_T + \frac{\Lambda_T \eta}{\sqrt{a}}.\end{aligned}$$

□

F.3 Proof of Theorem 5.2

Proof. Let R_T, S_T, S_0, and Λ_T be as in Lemma F.1, and set

$$L_T = \|\nabla f(0)\|_F + \frac{3\ell R_T^2}{2}.$$

The normalized update in Algorithm 3 gives $\|V_{t+1} - V_t\|_F \le \eta$, so all iterates and update segments lie in $\mathcal{B}_{R_T}$. Assumptions 3 and 4 imply that ∇f is ℓ-Lipschitz continuous. The local smoothness argument in the proof of Lemma 4.2, with $\rho = \ell$, therefore shows that J is L_T-smooth on this ball.

The descent argument in the proof of Lemma 4.4, with D_t and η in place of M_t and γ, gives

$$J(V_{t+1}) \le J(V_t) - \eta \|\nabla J(V_t)\|_F + 2\eta \|D_t - \nabla J(V_t)\|_F + \frac{L_T \eta^2}{2}.$$

Summing over $t = 0, \dots, T-1$, taking expectation, and using $J(V_T) \geq f_\star$, we obtain

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla J(V_t)\right\|_F &\leq \frac{\Delta}{\eta T} + \frac{2}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|D_t - \nabla J(V_t)\right\|_F + \frac{L_T\eta}{2} \\ &\leq \frac{\Delta}{\eta T} + \frac{2S_0}{aT} + 2\sqrt{2a}\,S_T + \frac{2\Lambda_T\eta}{\sqrt{a}} + \frac{L_T\eta}{2},\end{aligned} \tag{36}$$

where the last inequality follows from Lemma F.1.

Algorithm 3 sets $a = T^{-2/3}$ and $\eta = T^{-5/6}$. Since $T \geq 1$, the radius satisfies

$$R_T = \left\|V_0\right\|_F + T^{1/6} \leq (1 + \left\|V_0\right\|_F)T^{1/6}.$$

Using $(x+y)^2 \leq 2x^2 + 2y^2$ in the definition of Λ_T, we further obtain

$$\begin{aligned}\Lambda_T &\leq \left(3\ell^2 R_T^4 + 4\left\|\nabla f(0)\right\|_F^2 + 2\sigma^2\right)^{1/2} \\ &\leq \sqrt{3}\,\ell R_T^2 + 2\left\|\nabla f(0)\right\|_F + \sqrt{2}\,\sigma \\ &\leq \left(\sqrt{3}\,\ell(1 + \left\|V_0\right\|_F)^2 + 2\left\|\nabla f(0)\right\|_F + \sqrt{2}\,\sigma\right)T^{1/3}.\end{aligned}$$

Substituting the choices of a and η into (36), and then applying these bounds and the definitions of S_0, S_T, and L_T, gives

$$\begin{aligned}&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla J(V_t)\right\|_F \\ &\leq \Delta T^{-1/6} + 2\sigma\left\|V_0\right\|_F T^{-1/3} + 2\sqrt{2}\,\sigma R_T T^{-1/3} + 2\Lambda_T T^{-1/2} \\ &\quad + \left(\frac{\left\|\nabla f(0)\right\|_F}{2} + \frac{3\ell R_T^2}{4}\right)T^{-5/6} \\ &\leq \Delta T^{-1/6} + 2\sigma\left\|V_0\right\|_F T^{-1/3} + 2\sqrt{2}\,\sigma(1 + \left\|V_0\right\|_F)T^{-1/6} \\ &\quad + 2\left(\sqrt{3}\,\ell(1 + \left\|V_0\right\|_F)^2 + 2\left\|\nabla f(0)\right\|_F + \sqrt{2}\,\sigma\right)T^{-1/6} \\ &\quad + \left(\frac{\left\|\nabla f(0)\right\|_F}{2} + \frac{3\ell(1 + \left\|V_0\right\|_F)^2}{4}\right)T^{-1/2} \\ &\leq 8\left(1 + \Delta + (1 + \left\|V_0\right\|_F)(\left\|\nabla f(0)\right\|_F + \sigma) + \ell(1 + \left\|V_0\right\|_F)^2\right)T^{-1/6}.\end{aligned} \tag{37}$$

The last inequality uses $T^{-1/3} \leq T^{-1/6}$ and $T^{-1/2} \leq T^{-1/6}$.

Since I is independent of the oracle samples and uniform on $\{0, \dots, T-1\}$, we have

$$\mathbb{E}\left\|\nabla J(V_I)\right\|_F = \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla J(V_t)\right\|_F.$$

Thus, choosing T as in (18), with a sufficiently large universal constant, makes this expectation at most ϵ. Algorithm 3 uses one stochastic oracle evaluation at initialization and two per subsequent iteration, for a total of $2T - 1$ evaluations. Its stochastic oracle complexity therefore has the same order as T. □

G Experimental Details

This appendix specifies the experimental protocol used in Section 6. Hyperparameters are selected on validation data; the test splits are not used. Unless explicitly noted, the tasks, architectures, and LoRA configurations follow Mu & Klabjan (2026). We fix each task and model configuration before tuning the optimizers.

G.1 Task and model configurations

Table 2 summarizes the principal hyperparameters for the four reported settings. Our c_{adapt}, c_{adapt2}, and c_{norm} correspond to α^{adapt}, α^{adapt2}, and α^{norm}, respectively, in Mu & Klabjan (2026). Panels

Table 2: Experiment hyperparameters.

Hyperparameter	Logistic	ResNet-18	TinyLlama small	TinyLlama large
Batch size	512	512	16	16
LoRA rank	4	20	32	32
Input size	512	$32 \times 32 \times 3$	512 tokens	512 tokens
Training horizon	60/1,000 epochs	250 epochs	500 steps	250 steps
c_{adapt}	1.0	–	–	–
c_{adapt2}	0.8	15	0.2	562.9130541
c_{norm}	0.06	0.1	0.02	0.07931165
LoRA-NSGDM (α, γ)	$(0.2, 0.1)$	$(0.1, 0.1)$	$(0.2, 0.1)$	$(0.5, 0.2)$
LoRA-STORM (a, η_0)	$(0.5, 0.1)$	–	–	–

(a) and (b) of Figure 1 use 60 epochs. Figure 2 reports 250 epochs of ResNet-18 training. The small- and large-initialization panels of Figure 3 use 500 and 250 steps, respectively. Each optimizer iteration uses one minibatch. The first LoRA-STORM update uses one factor-gradient evaluation. Each subsequent update uses two evaluations on the same fresh minibatch, one at the current factors and one at the preceding factors. Thus, T updates require $2T - 1$ stochastic-oracle calls.

Logistic regression on fixed ResNet-18 features. We use the 50,000-example torchvision CIFAR-10 training split and make a deterministic 45,000/5,000 training/validation split. Images are resized to 224×224, normalized by the ImageNet mean and standard deviation, and encoded once by the penultimate layer of torchvision's ImageNet-1K V1 ResNet-18. The resulting 512-dimensional tensors are cached and shared by every method. A randomly initialized linear classifier weight is frozen and augmented by the unscaled rank-4 update BA. We initialize A with PyTorch's Kaiming-uniform rule and $B = 0$, so every optimizer starts from the same predictor. Training uses shuffled minibatches, cross-entropy, and no data augmentation or weight decay. With 45,000 examples and batch size 512, an epoch contains 88 updates; the final minibatch is smaller than 512.

CIFAR-10 classification with convolutional LoRA. We use the `uoft-cs/cifar10` Hugging Face mirror and initialize a torchvision ResNet-18 from scratch following Mu & Klabjan (2026). Rank-20 adapters are applied to every `conv1` and `conv2` in the residual blocks, and the convolutional downsample projections. The classification head is trained in full rank. All remaining ResNet parameters, including BatchNorm affine parameters, are frozen. The model remains in evaluation mode during optimization. This freezes the BatchNorm running statistics but leaves gradient computation enabled for the LoRA factors and classifier. Because the model is initialized from scratch and we perform no BatchNorm calibration pass, the stored running means and variances remain at their initialization. This evaluation-mode protocol is applied identically to all methods. Inputs are converted to tensors and normalized channelwise by mean and standard deviation 0.5, without augmentation. The deterministic 45,000/5,000 split is shared by all methods, the official test split is unused, and 250 epochs correspond to 22,000 updates.

TinyLlama instruction tuning. We use TinyLlama-1.1B-Chat-v1.0 and Alpaca. We reserve 2,000 examples for validation. Tuning evaluates a fixed 256-example subset, while reported terminal validation uses all 2,000 examples. Each example is rendered with the standard Alpaca instruction, optional input, and response fields, then truncated or padded to 512 tokens. Padding uses the end-of-sequence token and is masked from the causal-language-model loss.

We adapt only `q_proj`, `k_proj`, `v_proj`, and `o_proj`. The rank and LoRA scaling parameter are both 32, so the adapter multiplier is one; biases remain frozen. The configured LoRA dropout is 0.05, but the model is kept in evaluation mode and hence dropout is inactive. The A factor is initialized as a Gaussian random matrix with standard deviation $\sigma = 10^{-3}$ in the small initialization setting, or $\sigma = 1/r = 0.03125$ in the large initialization setting. In both settings, we set $B = 0$ initially. Each update uses one minibatch of 16 examples.

G.2 Baselines and selected hyperparameters

For coefficient c, the LoRA-GD stepsize rules used for LoRA-GD algorithms are

$$\eta_t^{\text{adapt}} = \min\left\{\frac{c}{\|V_t\|_F^2 + \|h_t\|_F}, 1\right\},$$
$$\eta_t^{\text{adapt2}} = \min\left\{\frac{c}{\|V_t\|_F^2 + \sqrt{\ell_t}}, 1\right\},$$
$$\eta_t^{\text{norm}} = \min\left\{\frac{c}{\sqrt{\|g_t\|_F}}, 1\right\}.$$

where V_t is the pair of LoRA factors and ℓ_t is the current minibatch loss. The quantities g_t and h_t are the gradients with respect to the factors and the product BA, respectively. The η_t^{adapt} rule is used only for logistic regression, as in Mu & Klabjan (2026). Consistent with Section 6, we also evaluate LoRA-STORM only for logistic regression because it uses two stochastic-oracle evaluations per iteration. Table 2 records every coefficient used in Figures 1–3. The fixed LoRA-GD learning rates $\{0.03, 0.05, 0.07\}$ for logistic regression and $\{0.03, 0.04, 0.045\}$ for convolutional ResNet-18 were reproduced and retained in the artifact bundle, but are omitted from the main comparison for the reason stated in Section 6. Hyperparameters are selected on validation data, and are ranked by training-loss AUC unless stated otherwise. Terminal validation loss is used as the tie breaker.

LoRA-NSGDM calibration. For every task we use the same coarse grid

$$\alpha \in \{0.05, 0.1, 0.2, 0.5, 1.0\}, \qquad \gamma \in \{0.01, 0.02, 0.05, 0.1, 0.2\}.$$

For logistic regression, all 25 candidates are trained for 5,280 steps and ranked by epoch-loss AUC, selecting $(0.2, 0.1)$. For convolutional ResNet-18, candidates are trained for 50 epochs. We minimize batch-loss AUC and select $(0.1, 0.1)$. TinyLlama uses a training budget of 500 steps. The selected parameters are $(0.2, 0.1)$ for small initialization and $(0.5, 0.2)$ for large initialization.

Large-initialization LoRA-GD calibration. In the large-initialization setting, the hyperparameters for the LoRA-GD algorithms are not specified in the open-source code repository of Mu & Klabjan (2026). Moreover, we cannot reuse the hyperparameters for small initialization because large initialization substantially changes the factor norm in the denominator of the LoRA-GD stepsize rules. We therefore tune a coarse grid of target initial effective stepsizes $\eta_0 \in \{0.05, 0.10, 0.20\}$. A fixed pilot minibatch gives $\ell_0 = 1.9944235$, $\|V_0\|_F = 75.0181198$, and $\|g_0\|_F = 2.5161352$. By the stepsize rules, we have

$$c_{\text{adapt2}} \in \{281.4565270, 562.9130541, 1125.8261081\},$$
$$c_{\text{norm}} \in \{0.07931165, 0.15862330, 0.31724661\}.$$

We select 562.9130541 and 0.07931165, respectively. Thus, the chosen η_t^{adapt2} rule targets pilot $\eta_0 = 0.10$, while the η_t^{norm} rule targets pilot $\eta_0 = 0.05$. The non-round coefficients follow algebraically from the pilot quantities above.

LoRA-STORM calibration. We use the cosine-decayed normalized stepsize

$$\eta_t = \eta_0\left[r_{\min} + \frac{1 - r_{\min}}{2}\left(1 + \cos\left(\frac{\pi t}{T-1}\right)\right)\right], \qquad t = 0, \ldots, T-1,$$

which decreases from η_0 to $r_{\min}\eta_0$ over a run of T steps. We fix $r_{\min} = 0.001$. The tuning grid is

$$a \in \{0.1, 0.2, 0.5, 1\},$$
$$\eta_0 \in \{0.1, 0.2, 0.5, 1\}.$$

We select $(a, \eta_0) = (0.5, 0.1)$. Because the schedule depends on T, the 5,280- and 88,000-step trajectories are run separately using the same selected hyperparameters.

G.3 Evaluation and plotting

The loss panels report pre-update minibatch training loss. Figure 1 uses moving-average windows 10, 200, and 200 in panels (a)–(c), respectively. Figure 2 uses window 50 in all three panels; panels (b) and (c) report the actual factor-update norm and minibatch-gradient norm. Figure 3 uses window 10 in both panels. Its curves prepend the common pre-update loss at step zero.

G.4 Hardware and software

The logistic-regression experiments were run on an Apple M2 CPU under macOS 15.0.1. The convolutional ResNet-18 experiments were run on an AMD EPYC 7542 server with one NVIDIA GeForce RTX 4090 GPU (24 GB). The TinyLlama experiments were run on an Intel Xeon Gold 6530 server with two NVIDIA GeForce RTX 4090 GPUs (48 GB each). Both GPU servers use Ubuntu 22.04.5 and CUDA 13.0. Each training trajectory uses one GPU; multiple GPUs are used only to execute independent trajectories concurrently.