FLAME: FIRST-MOMENT-FREE LOW-RANK GRADIENT PROJECTION WITH ADAM-STYLE CONVERGENCE GUARANTEES

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-rank gradient projection has emerged as a promising approach to optimizer-state compression, but existing methods still suffer from unstable optimization and high computational overhead. To understand these limitations, we develop a unified analytical framework and prove that, under independent random projections, first-order momentum retains only within-period history whereas the second-order state contains information from all periods, and that the induced adaptive updates effectively preserve direction in expectation. Guided by these findings, we propose FLAME (First-Moment-Free Low-Rank Adaptive Moment Estimation), which updates parameters using the current Gaussian-projected gradient normalized by the previous low-rank second-order estimate. It enables effective optimization while halving optimizer-state storage relative to existing Adam-based low-rank gradient projection methods and substantially reducing computational overhead. We prove an $\mathcal{O}(T^{-1/2})$ convergence rate for FLAME under discrete-time Adam-style optimization. FLAME further incorporates activation compression to reduce training memory. Experiments on LLaMA-3 fine-tuning and pretraining demonstrate competitive task performance and favorable convergence behavior. In terms of system efficiency, FLAME achieves the lowest peak memory among competing low-rank gradient-projection methods, reducing memory by 32–53% relative to full-rank Adam, while improving throughput by 23–45% over the fastest competitor.

1 INTRODUCTION

Large language models (LLMs) have achieved remarkable progress across a wide range of tasks, but their growing parameter counts make training increasingly memory-intensive. A major source of this cost is optimizer-state memory: Adam-style optimizers maintain parameter-sized first- and second-order moment estimates throughout training (Kingma & Ba, 2014; Reddi et al., 2018). To reduce this overhead, LoRA (Hu et al., 2022) and its variants (Lialin et al., 2024; Han et al., 2024) lower fine-tuning memory through low-rank parameterizations, but the resulting restriction can leave a performance gap relative to full-parameter fine-tuning (Xia et al., 2024).

Recently, low-rank gradient projection has emerged as a promising approach to reducing optimizer-state memory (Zhao et al., 2024; Hao et al., 2024; Zhu et al., 2025). Specifically, these methods project matrix-valued gradients into low-dimensional subspaces, maintain optimizer states and compute adaptive updates in the compressed space, and map the resulting updates back to the full parameter space. This approach has the potential to keep optimizer states low-rank while preserving the performance of full-parameter training. This paradigm has attracted considerable attention, motivating subsequent work to improve optimization through refined projection strategies (Zhang et al., 2025; He et al., 2025) or error-feedback mechanisms (Chen et al., 2025; Robert et al., 2025).

Despite these advances, existing methods still face challenges in ensuring optimization stability (He et al., 2025) and reducing computational overhead (Zhu et al., 2025). Theoretical understanding also remains limited: existing analyses primarily concern SGD-type methods or continuous-time formulations, leaving discrete-time Adam-style low-rank gradient projection largely unaddressed. To understand these limitations, we cast existing low-rank gradient projection methods into a unified an-

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.

alytical framework. We first analyze a central issue in low-rank gradient projection: optimizer-state information accumulated in previous subspaces may not be preserved across subspace transitions, potentially causing optimization instability (Liang et al., 2024; Xiao et al., 2025). We show that, in expectation under independent random projections (Zhu et al., 2025; He et al., 2025), first-order momentum contains only current-period information, whereas the second-order state accumulates information across periods. We further characterize the actual adaptive update induced by low-rank gradient projection and show that, under independent random projections, its expectation largely preserves the direction of the within-period momentum, while the second-moment state primarily controls its row-wise magnitude, thereby helping to explain the effectiveness of random projection for adaptive optimization. We also analyze the trade-off associated with the subspace refresh period.

Guided by these findings, we propose FLAME (First-Moment-Free Low-Rank Adaptive Moment Estimation), which eliminates first-order momentum, thereby avoiding the need to transfer momentum across refreshed subspaces, and adaptively scales the current Gaussian-projected gradient using the previous low-rank second-moment estimate. FLAME further compresses linear-layer activations, enabling projected weight gradients to be computed directly during backpropagation. Together, these designs enable effective optimization while halving optimizer-state storage relative to other low-rank gradient projection methods, reducing activation and gradient memory, and substantially lowering computational overhead. We establish an $\mathcal{O}(T^{-1/2})$ convergence rate for FLAME under discrete-time Adam-style optimization.

Experiments on LLaMA-3 models across fine-tuning and pretraining tasks show that FLAME delivers competitive task performance while achieving the lowest peak memory and highest training throughput among the competing low-rank gradient-projection methods.

Our contributions are summarized as follows:

1. We develop a unified analytical framework for the low-rank gradient projection paradigm.
2. We propose FLAME, a memory- and computation-efficient low-rank gradient projection method, and establish its convergence under discrete-time Adam-style optimization.
3. Extensive experiments on LLaMA-3 models validate FLAME's optimization effectiveness and system efficiency in both fine-tuning and pretraining.

2 Related Work

Low-Rank Gradient Projection. In this section, we discuss low-rank gradient projection methods. Other memory-efficient approaches, such as parameter-efficient fine-tuning, are deferred to Appendix C. Low-rank gradient projection has recently emerged as a promising memory-efficient training approach that seeks to preserve the performance of full-parameter training while keeping optimizer states low-rank. GaLore (Zhao et al., 2024) periodically recomputes a low-rank subspace from the gradient via truncated SVD, reuses it between refreshes for Adam-style updates in the projected space, and maps the updates back to the full parameter space. Flora (Hao et al., 2024) proposes a similar approach for Adafactor. APOLLO (Zhu et al., 2025) maintains optimizer states low-rank and estimates channel-wise gradient scaling factors for parameter updates. These methods inspired numerous follow-up works (Muhamed et al., 2024; Xiao et al., 2025; Shen et al., 2026).

Establishing convergence guarantees for low-rank gradient projection is challenging, and existing theory remains limited. GaLore provides a convergence analysis only for fixed projections under restrictive assumptions. GoLore (He et al., 2025) and I3S (Zhang et al., 2025) further show that GaLore can fail to converge in stochastic settings and propose random-projection variants with new convergence proofs. Liang et al. (2024) establish convergence for arbitrary projections, covering Adam-style optimizers, but in continuous time within a Hamiltonian-descent framework. Overall, these results are limited to SGD-type methods or continuous-time formulations, leaving discrete-time Adam-style optimization, the setting used in practice, largely unaddressed. LDAdam (Robert et al., 2025) develops a method with an AMSGrad-type (Reddi et al., 2018) convergence rate. However, its analysis relies on an additional contractive-projection assumption and is tailored to a specific error-feedback mechanism, offering only indirect guidance for general projection schemes.

Despite their promise, GaLore-style methods still face limitations in convergence behavior and optimization performance (He et al., 2025; Zhu et al., 2025), which recent work seeks to address

through refined projection strategies (Zhang et al., 2025; He et al., 2025) and error-feedback mechanisms (Chen et al., 2025; Robert et al., 2025). However, these methods do not fully address these limitations. To address these limitations, we develop a unified analytical framework for low-rank gradient projection and propose FLAME with convergence guarantees under discrete-time Adam-style settings while additionally reducing memory and computational overhead.

Convergence of Adam. Adam-style optimizers are the de facto choice for LLM training (Brown et al., 2020; Grattafiori et al., 2024), yet their convergence is theoretically delicate. Reddi et al. (2018) show that the original Adam can fail to converge even on simple convex problems. Subsequent analyses (Chen et al., 2019; Défossez et al., 2022) identify additional conditions required for Adam-type convergence, such as bounded gradients or restricted hyperparameter ranges. More relevant to this paper is ADOPT (Taniguchi et al., 2024), which removes the normalization factor's dependence on the current gradient and simplifies the convergence analysis by normalizing the current gradient using the previous second-moment estimate.

3 Analysis of Low-Rank Gradient Projection

3.1 Preliminaries

Adam. (Kingma & Ba, 2014) Let f be the objective function to be minimized, and $\mathbf{X}_t$ be the weight at step t with gradient $\mathbf{G}_t = \nabla f(\mathbf{X}_t)$. Adam tracks the first- and second-order moments as $\mathbf{M}_t = \beta_1 \mathbf{M}_{t-1} + (1-\beta_1)\mathbf{G}_t$ and $\mathbf{V}_t = \beta_2 \mathbf{V}_{t-1} + (1-\beta_2)\mathbf{G}_t^2$, respectively, where $\beta_1, \beta_2 \in (0,1)$, and we initialize both the first- and second-order moments at zero. It then computes the bias-corrected moments $\hat{\mathbf{M}}_t = \mathbf{M}_t/(1-\beta_1^t)$ and $\hat{\mathbf{V}}_t = \mathbf{V}_t/(1-\beta_2^t)$. Finally, the weights are updated as $\mathbf{X}_{t+1} = \mathbf{X}_t - \alpha \frac{\hat{\mathbf{M}}_t}{\sqrt{\hat{\mathbf{V}}_t + \epsilon}}$, where α is the learning rate, $\epsilon > 0$ is a small constant for numerical stability, and all the computations are applied element-wise. Adam needs to maintain the optimizer states $\mathbf{M}_t$ and $\mathbf{V}_t$, both of which have the same size as $\mathbf{X}_t$. This memory overhead is what low-rank gradient projection methods aim to reduce.

Low-rank gradient projection. Low-rank gradient projection (Zhao et al., 2024; Chen et al., 2025; Robert et al., 2025) reduces the optimizer-state memory by projecting gradients and maintaining the Adam moments in a low-dimensional subspace. In the following, we describe the family of low-rank gradient projection methods in a unified framework. Different methods vary in projection construction, refresh frequency, and optimizer-state handling across subspace transitions.

Consider a linear module with weight $\mathbf{X}_t \in \mathbb{R}^{m\times n}$ and gradient $\mathbf{G}_t$ at step t. Low-rank gradient projection chooses a projection matrix $\mathbf{\Omega}_t \in \mathbb{R}^{n\times r}$, such as an independent Gaussian projection, where r is the projection rank (typically $r \ll n$). The gradient is then projected as $\mathbf{U}_t = \mathbf{G}_t\mathbf{\Omega}_t \in \mathbb{R}^{m\times r}$, where $\mathbf{\Omega}_t$ is refreshed every τ steps. Let t_0 be the first step of the refresh period containing t (i.e., $t_0 = \tau\lfloor (t-1)/\tau \rfloor + 1$). The Adam moments are maintained in the projected space, as: $\mathbf{M}_t = \beta_1 \mathbf{M}_{t-1} + (1-\beta_1)\mathbf{U}_t$ and $\mathbf{V}_t = \beta_2 \mathbf{V}_{t-1} + (1-\beta_2)\mathbf{U}_t^{\odot 2}$. After bias correction, the update is computed element-wise and projected back to the original space: $\mathbf{X}_{t+1} = \mathbf{X}_t - \alpha \frac{\hat{\mathbf{M}}_t}{\sqrt{\hat{\mathbf{V}}_t + \epsilon}} \mathbf{\Omega}_t^\top$. For each weight matrix, this reduces the storage of optimizer states from $2mn$ to $2mr$.

Row-wise Analysis. We adopt a *row-wise* perspective to analyze the projected Adam update. For a matrix $\mathbf{A}$ (resp. a time-indexed matrix $\mathbf{A}_t$), we use $\mathbf{A}_a$ (resp. $\mathbf{A}_{t,a}$) to denote its a-th row. For clarity, at step t, we omit bias correction and focus on the core projected-update term $\frac{\mathbf{M}_t}{\sqrt{\mathbf{V}_t + \epsilon}} \mathbf{\Omega}_t^\top$. The corresponding update of $\mathbf{X}_{t,a}$ is given by $\left[\frac{\mathbf{M}_t}{\sqrt{\mathbf{V}_t + \epsilon}} \mathbf{\Omega}_t^\top\right]_a = \frac{\mathbf{M}_{t,a}}{\sqrt{\mathbf{V}_{t,a} + \epsilon}} \mathbf{\Omega}_t^\top = \mathbf{M}_{t,a} \mathbf{\Lambda}_t^a \mathbf{\Omega}_t^\top$, where $\mathbf{\Lambda}_t^a = \mathrm{diag}\big((\mathbf{V}_{t,a} + \epsilon)^{-1/2}\big) \in \mathbb{R}^{r\times r}$. Unrolling the first-moment recursion yields

$$\mathbf{M}_{t,a} \mathbf{\Lambda}_t^a \mathbf{\Omega}_t^\top = (1-\beta_1) \textstyle\sum_{s=1}^{t} \beta_1^{t-s} \mathbf{G}_{s,a} \mathbf{\Omega}_s \mathbf{\Lambda}_t^a \mathbf{\Omega}_t^\top. \tag{1}$$

Compared with the element-wise view, our row-wise perspective keeps the projection-induced matrix structure explicit while avoiding unwieldy tensor-level notation, providing a foundation for our theoretical analysis.

Challenges and Analysis Overview. Despite their memory savings, existing low-rank gradient projection methods still face fundamental challenges. They can suffer from unstable optimization (He

et al., 2025) and substantial computational overhead (Zhu et al., 2025). We therefore provide a unified theoretical and empirical analysis, and then use the obtained insights to guide the design of a method that addresses these limitations.

3.2 Can Optimizer Moments Be Preserved across Subspace Transitions?

In low-rank gradient projection, refreshing the projection subspace is necessary to avoid confining optimization to a fixed subspace. Prior work suggests that optimizer-state information from previous subspaces may not be preserved across transitions, particularly under random projections, raising concerns about optimization instability (Liang et al., 2024; Xiao et al., 2025). Yet, how optimizer moments behave across subspace transitions remains theoretically unclear. In the following, we analyze this question. First, we introduce a common assumption on random projection matrices.

Assumption 1 *The projection matrices $\{\mathbf{\Omega}_t \in \mathbb{R}^{n\times r}\}$ satisfy the following conditions:*

(i) ***Independence.*** *Projections at different refresh steps are mutually independent, and at each step t, $\mathbf{\Omega}_t \perp\!\!\!\perp \{\mathbf{G}_i\}_{i=1}^t$.*

(ii) ***Distribution.*** *$\mathbf{\Omega}_t$ is either $\sqrt{n/r}\,\mathbf{Q}$, where $\mathbf{Q}$ is sampled uniformly from the Stiefel manifold $\mathrm{St}(n,r)$ (Edelman et al., 1998), or a Gaussian matrix with i.i.d. $\mathcal{N}(0,1/r)$ entries.*

Assumption 1 covers the two RP schemes used in existing methods: Gaussian projections in Flora and APOLLO (Hao et al., 2024; Zhu et al., 2025) and orthogonal projections in GoLore (He et al., 2025). We next present Theorem 2. All proofs are in Appendix B.

Theorem 2 *Under Assumption 1, the following hold in expectation over the projections, with the gradients fixed:*

(a) ***First-order moment contains only current-period history:*** $\mathbb{E}\big[\mathbf{M}_t\mathbf{\Omega}_t^\top\big] = (1-\beta_1)\sum_{i=t_0}^{t}\beta_1^{t-i}\mathbf{G}_i,\ \forall\, t.$

(b) ***Second-order information accumulates across all periods in expectation:*** $\mathbb{E}\|\mathbf{V}_{t,a}\|_1 = (1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}\|\mathbf{G}_{i,a}\|^2,\ \forall\, t, a.$

The above Theorem reveals markedly different preservation behaviors for the two optimizer moments. The reconstructed first-order moment retains only the gradient history within the current refresh period, with contributions from previous periods vanishing in expectation. By contrast, the expected ℓ_1-norm of each row of $\mathbf{V}_t$ accumulates the exponentially weighted squared norms of the corresponding rows of all historical gradients. The results show that, in expectation, first-order information is limited to the current projection period, whereas second-order information accumulates across all periods.

3.3 Random Projection vs. SVD: Why Does a Crude Approximation Work?

The choice of the projection matrix $\mathbf{\Omega}_t$ is a key design decision in low-rank optimizer compression. SVD-based subspace selection is widely adopted in GaLore and subsequent methods (Zhao et al., 2024; Xiao et al., 2025; Chen et al., 2025), motivated by its optimal rank-r gradient approximation (Mirsky, 1960). However, such methods can be misled by biased gradient noise in low-SNR regimes (He et al., 2025) and exhibit a frozen dominant subspace, where the selected top singular subspace stops changing (Zhang et al., 2025), potentially leading to non-convergence under SGD.

On the other hand, random projection (RP) offers a computationally cheap alternative that constructs the projection matrix independently of the current gradient (Hao et al., 2024; Zhu et al., 2025; He et al., 2025). It can incur a large gradient reconstruction error (Wei et al., 2026). Prior work further suggests that the high variance of its gradient estimates may explain Flora's inferior performance (Muhamed et al., 2024). These limitations raise concerns about the use of RP in optimizer-state compression.

Despite these concerns, RP-based methods have demonstrated competitive performance in practice (Zhu et al., 2025; He et al., 2025). Motivated by this observation, we compare GaLore and GaLore_RP on GSM8K, CommonsenseQA, and CoLA, a subset of GLUE, across four projection

Table 1: Performance of GaLore and GaLore_RP on LLaMA-3 8B and LLaMA-3 3B across four different projection refresh periods. Detailed experimental settings are provided in Section 5.

Dataset	Method	LLaMA-3 8B				LLaMA-3 3B			
		100	200	300	400	100	200	300	400
CoLA	GaLore	**0.799**	**0.799**	**0.806**	**0.797**	0.777	0.776	0.777	0.777
	GaLore_RP	0.771	0.785	0.784	0.779	**0.782**	**0.784**	**0.783**	**0.782**
GSM8K	GaLore	0.638	0.644	**0.642**	**0.650**	0.373	0.372	0.376	0.372
	GaLore_RP	**0.649**	**0.654**	0.641	0.643	**0.426**	**0.412**	**0.411**	**0.416**
CommonsenseQA	GaLore	0.806	0.809	0.803	0.802	0.749	0.754	0.750	0.746
	GaLore_RP	**0.816**	**0.822**	**0.817**	**0.823**	**0.769**	**0.765**	**0.774**	**0.768**

refresh periods in Table 1. GaLore_RP replaces SVD with Gaussian RP at each projection update while keeping all others unchanged. GaLore_RP achieves competitive performance, raising the question of why its crude gradient approximation remains effective for parameter updates.

Importantly, optimizer-state compression is not merely a gradient reconstruction problem. The optimizer computes an adaptive update in the projected space and maps it to the parameter space, rather than applying the Adam update to a reconstructed gradient. This distinction calls for a new theoretical characterization of the actual parameter update. Continuing our row-wise analysis in Section 3.2, we study the expected parameter-space update $\mathbf{M}_{t,a}\mathbf{\Lambda}_t^a\mathbf{\Omega}_t^\top$ for each row a.

To facilitate the analysis, we first consider the factor $\mathbf{\Omega}_s\mathbf{\Lambda}\mathbf{\Omega}_t^\top$, treating the diagonal matrix $\mathbf{\Lambda}$ as fixed. Lemma 7 in Appendix B.2 shows that its conditional mean is zero when $s < t_0$, whereas for $s \geq t_0$ it reduces to $\mathrm{tr}(\mathbf{\Lambda})\mathbf{I}_n/r$. We next analyze the actual parameter update while accounting for the dependence of $\mathbf{\Lambda}$ on the current projector through the second-moment estimate.

Theorem 3 *Under Assumption 1, let $\mathbb{E}_t$ denote expectation over $\mathbf{\Omega}_t$, with the gradients fixed, and define the pre-refresh-period scaling $\bar{\mathbf{\Lambda}}_t^a = \mathrm{diag}\big((\beta_2^{t-t_0+1}\mathbf{V}_{t_0-1,a} + \epsilon)^{-1/2}\big)$. The row-wise update in (1) satisfies:*

$$\mathbb{E}_t\big[\mathbf{M}_{t,a}\mathbf{\Lambda}_t^a\mathbf{\Omega}_t^\top\big] = \frac{\mathrm{tr}(\bar{\mathbf{\Lambda}}_t^a)}{r}(1-\beta_1)\sum_{s=t_0}^{t}\beta_1^{t-s}\mathbf{G}_{s,a} + \mathbb{E}_t[\boldsymbol{\mathcal{R}}_{t,a}], \qquad \forall\, a, t,$$

where $\boldsymbol{\mathcal{R}}_{t,a} = \mathbf{M}_{t,a}(\mathbf{\Lambda}_t^a - \bar{\mathbf{\Lambda}}_t^a)\mathbf{\Omega}_t^\top$ is the remainder from replacing the current scaling with its pre-period counterpart. With $\sigma_{\min}(\cdot)$ denoting the smallest singular value, we have

$$\|\boldsymbol{\mathcal{R}}_{t,a}\|_2 \leq \frac{\|\mathbf{\Omega}_t\|_2}{\sigma_{\min}(\mathbf{\Omega}_t)}\max_{1\leq j\leq r}\frac{|(\mathbf{\Lambda}_t^a - \bar{\mathbf{\Lambda}}_t^a)_{jj}|}{(\bar{\mathbf{\Lambda}}_t^a)_{jj}}\,\|\mathbf{M}_{t,a}\bar{\mathbf{\Lambda}}_t^a\mathbf{\Omega}_t^\top\|_2, \qquad \textit{a.s.}$$

Theorem 3 decomposes the expected update into the expectation of a reference update using pre-period second-moment scaling and an expected remainder capturing the within-period contribution to this scaling. The inequality bounds the remainder relative to the reference update by two factors: $\|\mathbf{\Omega}_t\|_2/\sigma_{\min}(\mathbf{\Omega}_t)$ and $\max_{1\leq j\leq r}|(\mathbf{\Lambda}_t^a - \bar{\mathbf{\Lambda}}_t^a)_{jj}|/(\bar{\mathbf{\Lambda}}_t^a)_{jj}$. The first factor is $O(1)$ with high probability, while the second is typically controlled by $(t - t_0 + 1)(1-\beta_2)/\beta_2^{t-t_0+1}$. With the usual choice of $\beta_2 = 0.999$, the remainder therefore constitutes only a small fraction of the reference update. We provide a detailed theoretical analysis of both factors in Appendix B.4.

Therefore, the expected update for each row is approximately the first-order momentum accumulated within the current period scaled by a positive scalar determined by the second-moment state, thereby preserving its direction. This provides a theoretical explanation for RP's effectiveness and motivates its use as a computationally efficient choice for subspace selection that avoids the aforementioned SVD pitfalls.

3.4 How Often Should the Projection Subspace Be Updated?

Existing low-rank gradient projection methods often choose the refresh period empirically, with values ranging from one to several hundred steps (Robert et al., 2025; Zhao et al., 2024). Shorter refresh periods entail more frequent subspace transitions, which can induce optimizer-state misalignment (Chen et al., 2025) and loss of momentum information (Robert et al., 2025) (rigorously analyzed in Theorem 2), potentially disrupting the optimization dynamics. However, frequent refreshes diversify the subspaces of successive low-rank parameter updates, allowing their cumulative effect to better approximate full-space optimization. We next analyze this theoretically.

Proposition 4 *Under Assumption 1, consider a total of K steps, where K is a multiple of the refresh period τ. Let $\mathbf{C}_K(\tau) = K^{-1}\sum_{t=1}^{K}\mathbf{\Omega}_t\mathbf{\Omega}_t^\top$. For any $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$\|\mathbf{C}_K(\tau) - \mathbf{I}_n\|_2 \le 4(\log 9)\left[\sqrt{\frac{\tau q_\delta}{Kr}} + \frac{\tau q_\delta}{Kr}\right], \qquad q_\delta = n + \log(2/\delta),$$

Inspired by the classical random fusion-frame perspective (Bodmann, 2013), Proposition 4 uses $\|\mathbf{C}_K(\tau) - \mathbf{I}_n\|_2$ to quantify the discrepancy between cumulative subspace coverage $\mathbf{C}_K(\tau)$ and the full parameter space $\mathbf{I}_n$, showing that shorter refresh periods substantially reduce this discrepancy, whose bound grows at most linearly with τ. This reveals that the choice of refresh period entails an inherent trade-off between cumulative subspace coverage and momentum preservation. Comparable accuracy across the four refresh periods in Table 1 suggests no uniformly superior choice.

4 Method

Motivated by the preceding analysis, we propose **FLAME** (First-Moment-Free Low-Rank Adaptive Moment Estimation), a low-rank gradient projection method that co-designs adaptive optimization and the gradient-computation pipeline. As shown in Theorem 2, first-order history from previous projection periods contributes only zero-mean noise after reconstruction, naturally motivating the retention of only within-period gradients in the first-order moment. Under this scheme, the history window grows from one to τ gradients within each period, and the bias-correction exponent must increase accordingly to maintain proper normalization. However, this scheme can destabilize optimization and degrade training performance because each subspace refresh abruptly resets both the history window and the corresponding bias correction. To avoid this instability, FLAME directly uses the current projected gradient $\mathbf{U}_t$ for the parameter update. This prevents first-order history from previous subspaces from perturbing the current update and avoids abrupt changes at subspace refreshes, thereby stabilizing optimization. Crucially, FLAME requires no first-order optimizer state, halving optimizer-state storage relative to prior low-rank Adam methods.

FLAME uses a fresh Gaussian projection at every step ($\tau = 1$) for efficient subspace selection. For second-order adaptation at $t > 1$, following ADOPT (Taniguchi et al., 2024), we normalize the projected gradient $\mathbf{U}_t$ with the previous bias-corrected state $\hat{\mathbf{V}}_{t-1}$, decoupling the current gradient $\mathbf{G}_t$ from the adaptive second-moment estimate. These choices remove the dependence-induced remainder in Theorem 3, so the expected update uniformly scales the gradient within each row, facilitating convergence. Moreover, FLAME shifts the projection from optimizer-stage weight gradients to forward-pass activations, directly producing the required projected gradients without extra compression or altering upstream gradients, since $\mathbf{\Omega}_t$ is independent of the current gradient and low-rank projection is limited to linear-layer parameters, as established by AGO-CC (Wei et al., 2026). Algorithm 1 presents the FLAME workflow for a single linear layer.

Convergence Guarantee. We establish a convergence guarantee for FLAME's practical discrete-time Adam-style update. Assumption 5 comprises only conditions also used to establish Adam convergence (Défossez et al., 2022; Taniguchi et al., 2024). Our analysis mirrors practical low-rank training by modeling separate low-rank projections for individual linear-layer gradients together with uncompressed normalization-layer parameters. We present the convergence guarantee below.

Assumption 5 *Let $\boldsymbol{\theta}_t = (\mathbf{X}_t^1, \ldots, \mathbf{X}_t^p, \mathbf{X}_t^{\mathrm{norm}})$ denote all model parameters at step t. Here, $\mathbf{X}_t^l$ is the l-th linear-layer weight, and $\mathbf{X}_t^{\mathrm{norm}}$ stacks all normalization-layer weight vectors. Suppose that:*

(i) *f is a lower-bounded, L-smooth objective: $f(\boldsymbol{\theta}) \ge f_\star, \|\nabla f(\boldsymbol{\theta}) - \nabla f(\boldsymbol{\vartheta})\| \le L\|\boldsymbol{\theta} - \boldsymbol{\vartheta}\|$.*

Algorithm 1 FLAME update for a linear layer

Linear map: $\mathbf{A}_t \in \mathbb{R}^{s_t \times n} \mapsto \mathbf{Z}_t = \mathbf{A}_t \mathbf{X}_t^\top \in \mathbb{R}^{s_t \times m}$, with $\mathbf{X}_t \in \mathbb{R}^{m \times n}$
Input: Initial weight $\mathbf{X}_1$, total steps T, learning rate α, decay rate β_2, rank r, constant ϵ
Initialize second-order moments $\mathbf{V}_0, \hat{\mathbf{V}}_0 \in \mathbb{R}^{m \times r} \leftarrow 0$
for $t = 1, 2, \ldots, T$ **do**
  Forward Pass
  $\boldsymbol{\Omega}_t \in \mathbb{R}^{n \times r} \sim \mathcal{N}(0, \frac{1}{r})$ {sample random projection matrix}
  $\mathbf{Z}_t \leftarrow \mathbf{A}_t \mathbf{X}_t^\top \in \mathbb{R}^{s_t \times m}$ {linear-layer output}
  $\widetilde{\mathbf{A}}_t \leftarrow \mathbf{A}_t \, \boldsymbol{\Omega}_t \in \mathbb{R}^{s_t \times r}$ {store compressed activations and $\boldsymbol{\Omega}_t^\top$}
  Backward Pass
  $\mathbf{U}_t \leftarrow (\nabla_{\mathbf{Z}_t} \mathcal{L}_t)^\top \widetilde{\mathbf{A}}_t \in \mathbb{R}^{m \times r}$ {$\mathbf{U}_t = \mathbf{G}_t \boldsymbol{\Omega}_t$; without materializing $\mathbf{G}_t$}
  Pass $(\mathbf{U}_t, \boldsymbol{\Omega}_t^\top)$ to the optimizer
  Optimizer Update
  if $t > 1$ **then** $\mathbf{X}_{t+1} \leftarrow \mathbf{X}_t - \alpha \dfrac{\mathbf{U}_t}{\sqrt{\hat{\mathbf{V}}_{t-1} + \epsilon}} \boldsymbol{\Omega}_t^\top$ {no update at $t = 1$}
  $\mathbf{V}_t \leftarrow \beta_2 \mathbf{V}_{t-1} + (1 - \beta_2) \, \mathbf{U}_t^2$ {update second-order moment}
  $\hat{\mathbf{V}}_t \leftarrow \mathbf{V}_t / (1 - \beta_2^t)$ {bias correction for the next step}
end for
Output: $\mathbf{X}_{T+1}$

(ii) *The stochastic gradient $\mathbf{G}_t$ is unbiased with bounded conditional-mean norm and variance:*

$$\mathbb{E}[\mathbf{G}_t \mid \boldsymbol{\theta}_t] = \nabla f(\boldsymbol{\theta}_t), \qquad \|\mathbb{E}[\mathbf{G}_t \mid \boldsymbol{\theta}_t]\| \le R \quad a.s., \qquad \mathbb{E}\big[\|\mathbf{G}_t - \nabla f(\boldsymbol{\theta}_t)\|^2 \mid \boldsymbol{\theta}_t\big] \le \sigma^2.$$

Theorem 6 *Under Assumptions 5, for* $\alpha = \sqrt{2\epsilon C_0^2 \big(f(\boldsymbol{\theta}_1) - f_\star\big) / (LT(R^2 + \sigma^2) C_1)}$,

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \|\nabla f(\boldsymbol{\theta}_t)\|^2 \le \frac{\|\nabla f(\boldsymbol{\theta}_1)\|^2}{T} + \frac{\sqrt{2L(R^2 + \sigma^2) C_1 \big(f(\boldsymbol{\theta}_1) - f_\star\big)}}{\sqrt{\epsilon T}(1 - \sqrt{\beta_2})} = \mathcal{O}(T^{-1/2}),$$

with constants $C_0 := \sqrt{(R^2 + \sigma^2)(1 - \beta_2)} + \sqrt{\epsilon}(1 - \sqrt{\beta_2})$ *and* $C_1 := \big(\frac{n+r+1}{r}\big) C_0^2 + 8R\sqrt{\epsilon \big(\frac{n+r+1}{r}\big)(1 - \beta_2)}$.

Theorem 6 establishes an $O(T^{-1/2})$ convergence rate for FLAME, matching that of full-rank Adam (Taniguchi et al., 2024). This result highlights FLAME's strong convergence guarantee.

Advantages. By removing first-order momentum and jointly compressing linear-layer activations and weight gradients, FLAME requires only half the optimizer-state storage of other low-rank gradient projection methods while further reducing the memory footprints of activations and gradients. Crucially, these memory savings do not rely on a time-for-space trade-off: inexpensive Gaussian projections and first-moment-free updates reduce computational overhead, yielding higher training throughput. Beyond these efficiency gains, FLAME is designed in accordance with our theoretical analysis to induce an expected update that approximately applies a uniform scalar scaling within each gradient row, yielding effective optimization and favorable convergence properties.

5 Experiments

Experimental setup. We evaluate FLAME in both fine-tuning and pretraining settings. For fine-tuning, we use LLaMA-3 3B and 8B models (Grattafiori et al., 2024) on GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), GLUE (Wang et al., 2018), and CommonsenseQA (CSQA) (Talmor et al., 2019). For pretraining, we train a LLaMA-3 1B model from scratch on C4 (Raffel et al., 2020). We compare FLAME with full-rank Adam, LoRA, and five low-rank gradient projection methods: GaLore (Zhao et al., 2024), APOLLO (Zhu et al., 2025), GoLore (He et al., 2025), Fira (Chen et al., 2025), and LDAdam (Robert et al., 2025). For fine-tuning, all low-rank methods, including LoRA, use ranks of 8 and 32 for the 3B and 8B models, respectively. For

pretraining, all gradient-projection methods use a rank of 512. For all gradient-projection methods, low-rank projection is applied to all linear layers in the attention and MLP blocks. All fine-tuning experiments are run for five epochs with a learning rate of 2×10^{-5}. The batch sizes for the 3B/8B models are $16/4$ on GSM8K, $4/2$ on MATH, and $96/24$ on both GLUE and CSQA. For pretraining, we use a learning rate of 5×10^{-4} and a batch size of 64 with eight gradient accumulation steps, yielding an effective batch size of 512. All experiments are conducted in BF16 on a single NVIDIA H800 GPU with a fixed random seed of 42.

5.1 Task Performance

We first evaluate FLAME's task performance and convergence behavior while using half the optimizer-state memory of competing low-rank gradient-projection methods.

Downstream performance. Table 2 reports fine-tuning results for the LLaMA-3 3B and 8B models, measured by test accuracy on GSM8K, MATH, and CSQA and by the average score across the evaluated GLUE tasks. FLAME performs particularly well on the generative reasoning benchmarks and also remains competitive on the classification-oriented benchmarks. Overall, FLAME ranks among the top two low-rank methods in six of the eight settings and stays close to full-rank Adam, demonstrating competitive downstream performance with further optimizer-state memory savings.

Convergence behavior. We use the training loss during GSM8K fine-tuning with LLaMA-3 3B to assess convergence behavior. As shown in Figure 1a, FLAME exhibits a smooth and stable decrease in training loss, maintaining the lowest loss among the competitors throughout most of training and remaining closest to full-rank Adam. These results demonstrate FLAME's favorable empirical convergence and corroborate the guarantee in Theorem 6.

Pretraining performance. Figure 1b compares FLAME with Adam and three representative pretraining competitors. Although FLAME starts with relatively high perplexity, it rapidly narrows the gap and becomes nearly indistinguishable from Adam and GaLore by step 10,000. This shows that FLAME has potential for effective pretraining despite eliminating first-order momentum.

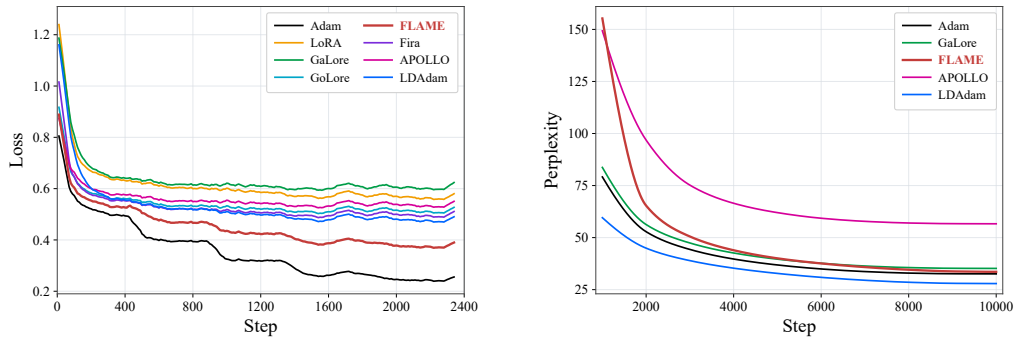
5.2 System Efficiency

We then evaluate whether FLAME's algorithmic design translates into practical system benefits by jointly delivering memory savings and higher training throughput on GSM8K and MATH.

Memory savings. Using peak allocated GPU memory as the primary metric, Table 3 shows that FLAME consistently achieves the lowest memory consumption among competitors. It reduces peak memory by 32–53% relative to full-rank Adam. We also report the estimated component-wise memory breakdown for each method in Appendix Tables 6 and 7. Together, FLAME achieves the largest memory savings among competitors by reducing activation, gradient, and optimizer-state memory.

Table 2: Fine-tuning performance of different methods on LLaMA-3 3B and LLaMA-3 8B across GSM8K, MATH, GLUE, and CSQA. Among FLAME and the competing low-rank methods, the best result is highlighted in bold and the second-best is underlined.

Methods	LLaMA-3 3B				LLaMA-3 8B			
	GSM8K	MATH	GLUE	CSQA	GSM8K	MATH	GLUE	CSQA
Adam	0.466	0.124	0.840	0.787	0.612	0.211	0.806	0.822
LoRA	0.396	0.113	0.805	0.764	0.614	0.210	0.806	0.825
FLAME	**0.410**	**0.122**	**0.795**	<u>0.767</u>	**0.654**	0.212	<u>0.816</u>	0.800
GaLore	0.376	0.108	0.727	0.754	0.650	**0.223**	0.773	0.809
APOLLO	0.333	0.101	0.757	0.701	0.574	0.211	0.799	0.789
GoLore	0.367	0.110	<u>0.791</u>	0.749	0.651	<u>0.222</u>	**0.828**	<u>0.830</u>
Fira	0.397	<u>0.119</u>	0.766	0.751	<u>0.652</u>	0.218	0.657	0.818
LDAdam	<u>0.407</u>	**0.122**	0.720	**0.779**	0.628	0.209	0.812	**0.836**



(a) GSM8K fine-tuning loss on LLaMA-3 3B.

(b) LLaMA-3 1B validation perplexity on C4.

Figure 1: Training loss during GSM8K fine-tuning with LLaMA-3 3B (left) and validation perplexity during C4 pretraining with LLaMA-3 1B (right).

Table 3: Peak allocated GPU memory (GB). Among the competing low-rank gradient-projection methods, the best result is highlighted in bold and the second-best is underlined.

Model	Dataset	Baselines		Low-rank gradient-projection methods					
		Adam	LoRA	FLAME	GaLore	APOLLO	GoLore	Fira	LDAdam
LLaMA-3 3B	GSM8K	57.67	47.88	**38.97**	<u>47.16</u>	47.78	47.20	47.20	47.20
	MATH	31.28	23.35	**19.68**	<u>20.77</u>	21.38	20.81	20.81	20.81
LLaMA-3 8B	GSM8K	62.89	35.33	**29.25**	36.52	37.07	36.52	36.52	<u>35.20</u>
	MATH	62.88	34.11	**29.97**	36.53	37.05	36.53	36.53	<u>35.20</u>

Training throughput. Table 4 shows that FLAME consistently achieves the highest throughput among competitors, outperforming the second-fastest method, APOLLO, by 23–45% owing to inexpensive Gaussian random projections and the elimination of first-moment updates.

Table 4: Training throughput (token/s). Among the competing low-rank gradient-projection methods, the best result is highlighted in bold and the second-best is underlined.

Model	Dataset	Baselines		Low-rank gradient-projection methods					
		Adam	LoRA	FLAME	GaLore	APOLLO	GoLore	Fira	LDAdam
LLaMA-3 3B	GSM8K	6,001	7,059	**7,438**	2,827	<u>6,029</u>	4,269	3,753	4,919
	MATH	4,430	6,250	**5,775**	1,617	<u>4,463</u>	2,427	1,916	3,429
LLaMA-3 8B	GSM8K	1,875	3,691	**3,004**	754	<u>2,071</u>	884	675	1,320
	MATH	1,487	3,214	**2,374**	562	<u>1,677</u>	664	498	1,038

6 Conclusion

In this paper, we developed a unified analytical framework for low-rank gradient projection, characterizing how optimizer-state history accumulates and how the resulting adaptive updates behave under random projections. Building on these theoretical insights, we proposed FLAME, a first-moment-free low-rank gradient projection method that achieves competitive optimization performance while reducing memory usage and computational overhead relative to existing low-rank gradient projection methods. We further established a discrete-time Adam-style convergence guarantee for FLAME. Experiments in both pretraining and fine-tuning on LLaMA-3 models validate FLAME's optimization effectiveness and demonstrate practical gains in system efficiency.

AI USE STATEMENT

In this work, we used generative AI tools solely to improve the clarity, grammar, and readability of the manuscript. We did not use generative AI tools for any other research tasks requiring disclosure under the ICLR policy. All AI-assisted revisions were reviewed by the authors to ensure that they accurately reflect the intended scientific meaning. We take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide the complete FLAME procedure in Algorithm 1 and detailed experimental settings in Section 5. The assumptions underlying our theoretical results are explicitly stated in the main text, and all proofs are provided in Appendix B. The source code is provided in the supplementary material.

REFERENCES

Bernhard G. Bodmann. Random fusion frames are nearly equiangular and tight. *Linear Algebra and its Applications*, 439(5):1401–1414, 2013.

Tom B. Brown et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33*, pp. 1877–1901, 2020.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *CoRR*, abs/1604.06174, 2016.

Xi Chen, Kaituo Feng, Changsheng Li, Xunhao Lai, Xiangyu Yue, Ye Yuan, and Guoren Wang. Fira: Can we achieve full-rank training of LLMs under low-rank constraint? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of A class of adam-type algorithms for non-convex optimization. In *7th International Conference on Learning Representations*, 2019.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021.

Alexandre Défossez, Léon Bottou, Francis R. Bach, and Nicolas Usunier. A simple convergence proof of adam and adagrad. *Transactions on Machine Learning Research*, 2022, 2022.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 8-bit optimizers via block-wise quantization. In *The Tenth International Conference on Learning Representations, ICLR 2022*, 2022.

Alan Edelman, Tomás A. Arias, and Steven Thomas Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *CoRR*, abs/2407.21783, 2024.

Andi Han, Jiaxiang Li, Wei Huang, Mingyi Hong, Akiko Takeda, Pratik Kumar Jawanpuria, and Bamdev Mishra. SLTrain: A sparse plus low-rank approach for parameter and memory efficient pretraining. In *Advances in Neural Information Processing Systems 37*, 2024.

Yongchang Hao, Yanshuai Cao, and Lili Mou. Flora: Low-rank adapters are secretly gradient compressors. In *Forty-first International Conference on Machine Learning*, pp. 17554–17571, 2024.

Yutong He, Pengrui Li, Yipeng Hu, Chuyan Chen, and Kun Yuan. Subspace optimization for large language models with convergence guarantees. In *Forty-second International Conference on Machine Learning*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2790–2799, 2019.

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 *The Tenth International Conference on Learning Representations*, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

Bingrui Li, Jianfei Chen, and Jun Zhu. Memory efficient optimizers with 4-bit states. In *Advances in Neural Information Processing Systems 36*, 2023.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pp. 4582–4597, 2021.

Vladislav Lialin, Sherin Muckatira, Namrata Shivagunde, and Anna Rumshisky. ReLoRA: High-rank training through low-rank updates. In *The Twelfth International Conference on Learning Representations*, 2024.

Kaizhao Liang, Bo Liu, Lizhang Chen, and Qiang Liu. Memory-efficient LLM training with online subspace descent. In *Advances in Neural Information Processing Systems*, 2024.

Ziyue Liu, Ruijie Zhang, Zhengyang Wang, Mingsong Yan, Zi Yang, Paul D. Hovland, Bogdan Nicolae, Franck Cappello, Sui Tang, and Zheng Zhang. CoLA: Compute-efficient pre-training of LLMs via low-rank activation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4627–4645, 2025.

L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 11:50–59, 1960.

Ionut-Vlad Modoranu, Mher Safaryan, Grigory Malinovsky, Eldar Kurtic, Thomas Robert, Peter Richtárik, and Dan Alistarh. Microadam: Accurate adaptive optimization with low space overhead and provable convergence. In *Advances in Neural Information Processing Systems*, 2024.

Aashiq Muhamed, Oscar Li, David P. Woodruff, Mona T. Diab, and Virginia Smith. GRASS: compute efficient low-memory LLM training with structured sparse gradients. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 14978–15003, 2024.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *6th International Conference on Learning Representations*, 2018.

Minsoo Rhu, Natalia Gimelshein, Jason Clemons, Arslan Zulfiqar, and Stephen W. Keckler. vDNN: Virtualized deep neural networks for scalable, memory-efficient neural network design. In *49th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 18:1–18:13, 2016.

Thomas Robert, Mher Safaryan, Ionut-Vlad Modoranu, and Dan Alistarh. LDAdam: Adaptive optimization from low-dimensional gradient statistics. In *The Thirteenth International Conference on Learning Representations*, 2025.

Charbel Sakr and Brucek Khailany. ESPACE: Dimensionality reduction of activations for model compression. In *Advances in Neural Information Processing Systems 37*, 2024.

Yara Shamshoum, Nitzan Hodos, Yuval Sieradzki, and Assaf Schuster. CompAct: Compressed activations for memory-efficient LLM training. In *Proceedings of the Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*, pp. 1511–1524, 2025.

Wei Shen, Zhang Yaxiang, Minhui Huang, Mengfan Xu, Jiawei Zhang, and Cong Shen. MLorc: Momentum low-rank compression for memory efficient large language model adaptation. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026.

Jiang-Xin Shi, Wen-Da Wei, Jin-Fei Qi, Xuanyu Chen, Tong Wei, and Yu-Feng Li. Memory-efficient fine-tuning via low-rank activation compression. *CoRR*, abs/2509.23472, 2025.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4149–4158, 2019.

Shohei Taniguchi, Keno Harada, Gouki Minegishi, Yuta Oshima, Seong Cheol Jeong, Go Nagahara, Tomoshi Iiyama, Masahiro Suzuki, Yusuke Iwasawa, and Yutaka Matsuo. ADOPT: modified adam can converge with any β_2 with the optimal rate. In *Advances in Neural Information Processing Systems 37*, 2024.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. DyLoRA: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 3266–3279, 2023.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In *Compressed Sensing*, pp. 210–268. 2012.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP*, pp. 353–355, 2018.

Wen-Da Wei, Han-Bin Fang, Yang-Di Liu, Jiang-Xin Shi, James Kwok, and Yufeng Li. Activation compression in llms: Theoretical analysis and efficient algorithm. *CoRR*, abs/2605.01255, 2026.

Wenhan Xia, Chengwei Qin, and Elad Hazan. Chain of LoRA: Efficient fine-tuning of language models via residual learning. *CoRR*, abs/2401.04151, 2024.

Jinqi Xiao, Shen Sang, Tiancheng Zhi, Jing Liu, Qing Yan, Linjie Luo, and Bo Yuan. COAP: memory-efficient training with correlation-aware gradient projection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition,*, pp. 30116–30126, 2025.

Haochen Zhang, Junze Yin, Guanchu Wang, Zirui Liu, Lin Yang, Tianyi Zhang, Anshumali Shrivastava, and Vladimir Braverman. Breaking the frozen subspace: Importance sampling for low-rank optimization in LLM pretraining. In *Advances in Neural Information Processing Systems 38*, 2025.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. GaLore: Memory-efficient LLM training by gradient low-rank projection. In *Forty-first International Conference on Machine Learning*, pp. 61121–61143, 2024.

Chao Zhou, Tom Jacobs, Advait Gadhikar, and Rebekka Burkholz. Pay attention to small weights. In *Advances in Neural Information Processing Systems 38*, pp. 162222–162257, 2025.

Hanqing Zhu, Zhenyu Zhang, Wenyan Cong, Xi Liu, Sem Park, Vikas Chandra, Bo Long, David Z. Pan, Zhangyang Wang, and Jinwon Lee. APOLLO: SGD-like memory, AdamW-level performance. In *Proceedings of Machine Learning and Systems*, 2025.

A Additional Experimental Results

A.1 Ablation Studies on First-Order Momentum and Projection Matrix

In this section, we conduct ablation studies on first-order momentum and projection-matrix construction using LLaMA-3 8B, comparing GSM8K accuracy across four configurations. We first describe the three ablation configurations—FLAME-Momentum, Gaussian RP, and Orthogonal RP—reported in Table 5. FLAME-Momentum maintains projected first-order momentum using only gradients from the current projection period. Within each period, the momentum history grows from one to τ gradients; accordingly, the bias-correction exponent increases from 1 to τ and is reset after each subspace refresh. In Table 5, we use $\beta_1 = 0.9$ and $\tau = 200$. Gaussian RP and Orthogonal RP both use $\beta_1 = 0.9$ with a refresh period of $\tau = 1$, without first-moment bias correction; they differ only in the construction of the projection matrix.

Comparing FLAME with FLAME-Momentum, we find that FLAME-Momentum, which retains a longer history of within-period gradients in the first-order moment, achieves an accuracy of 0.621, compared with 0.654 for FLAME. This degradation is consistent with the motivation in Section 4: each subspace refresh abruptly resets both the momentum history and its bias-correction counter, potentially disrupting the optimization dynamics.

Comparing the projection constructions, Gaussian RP substantially outperforms Orthogonal RP (0.575 vs. 0.514). Both Gaussian and orthogonal projections are isotropic in expectation, satisfying $\mathbb{E}[\mathbf{\Omega}\mathbf{\Omega}^\top] = \mathbf{I}_n$. However, their per-step geometries differ: scaled orthogonal RP has fixed singular values, whereas Gaussian RP exhibits random singular-value fluctuations. We conjecture that the additional per-step variability introduced by Gaussian projections may promote exploration of the optimization landscape and help avoid unfavorable regions.

Table 5: Ablation studies on first-order momentum and projection-matrix construction for LLaMA-3 8B fine-tuning on GSM8K.

Method	**FLAME**	**FLAME-Momentum**	**Gaussian RP**	**Orthogonal RP**
Accuracy	**0.654**	<u>0.621</u>	0.575	0.514

A.2 Component-Wise Memory Analysis

To provide a fine-grained view of FLAME's memory savings, Tables 6 and 7 report, for each method, the estimated memory consumption of model parameters, activations, gradients, and optimizer states for the 3B and 8B models, respectively. FLAME simultaneously reduces activation, gradient, and optimizer-state memory. Together, these complementary reductions enable FLAME to deliver the largest overall memory savings among the evaluated methods across model scales.

Table 6: Estimated memory breakdown for LLaMA-3 3B fine-tuning on GSM8K (GB).

Memory component	**Baselines**		**Low-rank gradient-projection methods**					
	Adam	**LoRA**	**FLAME**	**GaLore**	**APOLLO**	**GoLore**	**Fira**	**LDAdam**
Model	5.98	7.46	5.98	5.98	5.98	5.98	5.98	5.98
Activations	31.57	31.57	24.10	31.57	31.57	31.57	31.57	31.57
Gradients	6.03	1.48	0.75	6.03	6.03	6.03	6.03	6.03
Optimizer states	11.99	2.95	0.75	1.69	1.50	1.69	1.69	1.61

Table 7: Estimated memory breakdown for LLaMA-3 8B fine-tuning on GSM8K (GB). The LoRA configuration includes the trainable embedding and LM head.

Memory component	**Baselines**		**Low-rank gradient-projection methods**					
	Adam	**LoRA**	**FLAME**	**GaLore**	**APOLLO**	**GoLore**	**Fira**	**LDAdam**
Model	14.96	16.97	14.96	14.96	14.96	14.96	14.96	14.96
Activations	13.45	13.45	10.19	13.45	13.45	13.45	13.45	13.45
Gradients	14.96	2.01	2.07	14.96	14.96	14.96	14.96	14.96
Optimizer states	29.92	4.02	2.07	4.58	4.14	4.58	4.58	4.47

B PROOFS

At the outset of the proofs, we adopt the following norm notation for clarity: $\|\cdot\|$ denotes the Euclidean 2-norm for vectors and the Frobenius norm for matrices; the spectral norm of a matrix $\mathbf{A}$ is denoted by $\|\mathbf{A}\|_2$.

B.1 PROOF OF THEOREM 2

Throughout this proof the gradients are fixed, and the expectation is over all period projectors. We first record the projection moments used below. For either distribution in Assumption 1, writing $\mathbf{\Omega} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_r)$ gives

$$\mathbb{E}\mathbf{\Omega} = 0, \qquad \mathbb{E}[\boldsymbol{\omega}_j \boldsymbol{\omega}_j^\top] = \frac{1}{r}\mathbf{I}_n, \qquad \mathbb{E}[\mathbf{\Omega}\mathbf{\Omega}^\top] = \mathbf{I}_n. \tag{2}$$

For Gaussian projections, the entries are centered and independent, so $\mathbb{E}[\Omega_{ij}\Omega_{kj}] = 0$ for $i \neq k$, while $\mathbb{E}\Omega_{ij}^2 = 1/r$. For scaled Stiefel projections, $\mathbf{\Omega} = \sqrt{n/r}\,\mathbf{Q}$, invariance under a sign flip of any row gives zero means and $\mathbb{E}[Q_{ij}Q_{kj}] = 0$ for $i \neq k$. Coordinate symmetry and $\sum_{i=1}^n Q_{ij}^2 = 1$ give $\mathbb{E}Q_{ij}^2 = 1/n$; after scaling, $\mathbb{E}\Omega_{ij}^2 = 1/r$. Thus both distributions satisfy the first two identities. Summing $\mathbb{E}[\boldsymbol{\omega}_j \boldsymbol{\omega}_j^\top] = \mathbf{I}_n/r$ over $j = 1, \ldots, r$ gives the third.

The zero initialization and the moment recursions give

$$\begin{aligned} \mathbf{M}_t &= (1-\beta_1)\sum_{i=1}^t \beta_1^{t-i}\mathbf{G}_i\mathbf{\Omega}_i, \\ \mathbf{V}_t &= (1-\beta_2)\sum_{i=1}^t \beta_2^{t-i}(\mathbf{G}_i\mathbf{\Omega}_i)^{\odot 2}. \end{aligned} \tag{3}$$

(a) First-order moment. If $t_0 \leq i \leq t$, the same projector is used at steps i and t. If $i < t_0$, the two projectors belong to different periods and are independent. Consequently, by equation 2,

$$\mathbb{E}[\mathbf{\Omega}_i\mathbf{\Omega}_t^\top] = \begin{cases} \mathbf{I}_n, & t_0 \leq i \leq t, \\ \mathbf{0}, & i < t_0. \end{cases} \tag{4}$$

Multiplying the first identity in equation 3 by $\mathbf{\Omega}_t^\top$ and taking expectation yields

$$\mathbb{E}[\mathbf{M}_t\mathbf{\Omega}_t^\top] = (1-\beta_1)\sum_{i=t_0}^t \beta_1^{t-i}\mathbf{G}_i. \tag{5}$$

(b) Second-order moment. All entries of $\mathbf{V}_t$ are nonnegative. Thus the row-wise ℓ_1 norm is the sum of its entries, and

$$\|\mathbf{V}_{t,a}\|_1 = (1-\beta_2)\sum_{i=1}^t \beta_2^{t-i}\|\mathbf{G}_{i,a}\mathbf{\Omega}_i\|_2^2. \tag{6}$$

For each i, we have

$$\mathbb{E}\|\mathbf{G}_{i,a}\mathbf{\Omega}_i\|_2^2 = \mathbf{G}_{i,a}\mathbb{E}[\mathbf{\Omega}_i\mathbf{\Omega}_i^\top]\mathbf{G}_{i,a}^\top = \|\mathbf{G}_{i,a}\|_2^2. \tag{7}$$

Taking expectation in the preceding sum proves part (b). □

B.2 LEMMA 7 AND ITS PROOF

Lemma 7 *Under Assumption 1, let $\mathbb{E}_t$ denote expectation over $\mathbf{\Omega}_t$, keeping the gradients and projectors from previous periods fixed. For any fixed diagonal $\mathbf{\Lambda} \in \mathbb{R}^{r\times r}$,*

$$\mathbb{E}_t\left[\mathbf{\Omega}_s\mathbf{\Lambda}\mathbf{\Omega}_t^\top\right] = \mathbf{1}_{\{s\geq t_0\}}\,\frac{\operatorname{tr}(\mathbf{\Lambda})}{r}\mathbf{I}_n, \qquad s \leq t. \tag{8}$$

The expectation $\mathbb{E}_t$ fixes the gradients and all projectors from earlier periods. Write $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_r)$. For $s \geq t_0$, $\mathbf{\Omega}_s = \mathbf{\Omega}_t$; by equation 2,

$$\mathbb{E}_t[\mathbf{\Omega}_s \mathbf{\Lambda} \mathbf{\Omega}_t^\top] = \sum_{j=1}^{r} \lambda_j \mathbb{E}_t[\boldsymbol{\omega}_j \boldsymbol{\omega}_j^\top] = \frac{\mathrm{tr}\,\mathbf{\Lambda}}{r} \mathbf{I}_n. \tag{9}$$

For $s < t_0$, both $\mathbf{\Omega}_s$ and $\mathbf{\Lambda}$ are fixed under this expectation, while $\mathbb{E}_t \mathbf{\Omega}_t = 0$. Therefore

$$\mathbb{E}_t[\mathbf{\Omega}_s \mathbf{\Lambda} \mathbf{\Omega}_t^\top] = \mathbf{\Omega}_s \mathbf{\Lambda} \mathbb{E}_t[\mathbf{\Omega}_t^\top] = \mathbf{0}. \tag{10}$$

These two cases prove the identity for either projection distribution. □

B.3 Proof of Theorem 3

We first consider the reference update $\mathbf{Z}_{t,a}^0$. Let $h = t - t_0 + 1$, and use the reference scaling $\bar{\mathbf{\Lambda}}_t^a$ defined in the theorem. Since $\mathbf{V}_{t_0-1,a}$ depends only on previous-period projectors and fixed gradients, $\bar{\mathbf{\Lambda}}_t^a$ is fixed under $\mathbb{E}_t$. Define the reference update

$$\mathbf{Z}_{t,a}^0 := \mathbf{M}_{t,a} \bar{\mathbf{\Lambda}}_t^a \mathbf{\Omega}_t^\top. \tag{11}$$

Then, with the remainder defined in the theorem, we have

$$\mathbf{M}_{t,a} \mathbf{\Lambda}_t^a \mathbf{\Omega}_t^\top = \mathbf{Z}_{t,a}^0 + \boldsymbol{\mathcal{R}}_{t,a}. \tag{12}$$

Expanding $\mathbf{M}_{t,a}$ and applying Lemma 7 gives

$$\begin{aligned} \mathbb{E}_t \mathbf{Z}_{t,a}^0 &= (1-\beta_1) \sum_{s=1}^{t} \beta_1^{t-s} \mathbf{G}_{s,a} \mathbb{E}_t[\mathbf{\Omega}_s \bar{\mathbf{\Lambda}}_t^a \mathbf{\Omega}_t^\top] \\ &= \frac{\mathrm{tr}(\bar{\mathbf{\Lambda}}_t^a)}{r} (1-\beta_1) \sum_{s=t_0}^{t} \beta_1^{t-s} \mathbf{G}_{s,a}. \end{aligned} \tag{13}$$

Taking the conditional expectation of the decomposition gives the desired mean.

We next bound the remainder $\boldsymbol{\mathcal{R}}_{t,a}$. We define

$$\begin{aligned} &\mathbf{x} := \mathbf{M}_{t,a} \bar{\mathbf{\Lambda}}_t^a, \\ &\mathbf{E} := (\bar{\mathbf{\Lambda}}_t^a)^{-1} (\mathbf{\Lambda}_t^a - \bar{\mathbf{\Lambda}}_t^a), \qquad \theta_{t,a} := \|\mathbf{E}\|_2. \end{aligned} \tag{14}$$

The inverse exists since $\epsilon > 0$. Both scaling matrices are diagonal, so $\theta_{t,a}$ is precisely the maximum relative diagonal difference in the theorem. We have

$$\begin{aligned} \|\boldsymbol{\mathcal{R}}_{t,a}\|_2 &= \|\mathbf{x}\mathbf{E}\mathbf{\Omega}_t^\top\|_2 \leq \|\mathbf{x}\|_2 \theta_{t,a} \|\mathbf{\Omega}_t\|_2, \\ \|\mathbf{Z}_{t,a}^0\|_2 &= \|\mathbf{x}\mathbf{\Omega}_t^\top\|_2 \geq \sigma_{\min}(\mathbf{\Omega}_t) \|\mathbf{x}\|_2. \end{aligned} \tag{15}$$

The smallest singular value is positive almost surely. For scaled Stiefel projections this follows from $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_r$. For Gaussian projections, the first r rows form a square Gaussian matrix, whose determinant is nonzero almost surely. Combining the two inequalities proves the bound, including $\mathbf{x} = 0$, when both updates vanish. □

B.4 Analysis of the Remainder Bound Factors

We state and prove the two supporting lemmas used to interpret the bound in Theorem 3.

Lemma 8 (Gaussian projector conditioning) *Let $\mathbf{\Omega} \in \mathbb{R}^{n \times r}$ have independent $\mathcal{N}(0, 1/r)$ entries. Suppose $r \leq \alpha n$ for a fixed $\alpha \in (0,1)$. With probability at least $1 - 2\exp(-n(1-\sqrt{\alpha})^2/8)$, the matrix has full column rank and its spectral condition number satisfies*

$$\kappa(\mathbf{\Omega}) := \frac{\sigma_{\max}(\mathbf{\Omega})}{\sigma_{\min}(\mathbf{\Omega})} = \frac{\|\mathbf{\Omega}\|_2}{\sigma_{\min}(\mathbf{\Omega})} \leq \frac{3+\sqrt{\alpha}}{1-\sqrt{\alpha}} = O_\alpha(1). \tag{16}$$

B.4.1 Proof of Lemma 8

For $\mathbf{A} := \sqrt{r}\,\mathbf{\Omega}$, Theorem 5.32 and Proposition 5.34 of Vershynin (2012) (see also Corollary 5.35 therein) give, for any $u > 0$,

$$\begin{aligned} \mathbb{P}\{\sigma_{\max}(\mathbf{A}) > \sqrt{n} + \sqrt{r} + u\} &\le e^{-u^2/2}, \\ \mathbb{P}\{\sigma_{\min}(\mathbf{A}) < \sqrt{n} - \sqrt{r} - u\} &\le e^{-u^2/2}. \end{aligned} \tag{17}$$

Choose $u = (1 - \sqrt{\alpha})\sqrt{n}/2$. A union bound and $r \le \alpha n$ imply that, with the stated probability,

$$\sigma_{\min}(\mathbf{A}) \ge \frac{1 - \sqrt{\alpha}}{2}\sqrt{n} > 0, \qquad \sigma_{\max}(\mathbf{A}) \le \frac{3 + \sqrt{\alpha}}{2}\sqrt{n}. \tag{18}$$

Taking their ratio proves the result because scalar multiplication leaves the condition number unchanged. □

Lemma 9 (Denominator perturbation and second-moment decay) *Use the notation of Theorem 3, and let $h = t - t_0 + 1$ and $\theta_{t,a} := \max_j |(\mathbf{\Lambda}_t^a - \bar{\mathbf{\Lambda}}_t^a)_{jj}|/(\bar{\mathbf{\Lambda}}_t^a)_{jj}$. Suppose that, for some $C > 0$ independent of β_2 and h,*

$$(\mathbf{U}_{q,a})_j^2 \le C((\mathbf{V}_{t_0-1})_{aj} + \epsilon), \qquad t_0 \le q \le t, \quad 1 \le j \le r. \tag{19}$$

Then, deterministically,

$$\theta_{t,a} \le \frac{C}{2}\frac{1 - \beta_2^h}{\beta_2^h} \le \frac{Ch(1 - \beta_2)}{2\beta_2^h}. \tag{20}$$

B.4.2 Proof of Lemma 9 and Numerical Analysis

Expanding the second-moment recursion within the current period gives

$$\mathbf{V}_{t,a} = \beta_2^h \mathbf{V}_{t_0-1,a} + \mathbf{d}_{t,a}, \qquad \mathbf{d}_{t,a} := (1 - \beta_2)\sum_{q=t_0}^{t} \beta_2^{t-q}(\mathbf{U}_{q,a})^{\odot 2}. \tag{21}$$

For each coordinate, set $b_j := \beta_2^h(\mathbf{V}_{t_0-1})_{aj} + \epsilon > 0$ and $\rho_j := (\mathbf{d}_{t,a})_j/b_j \ge 0$. Then

$$\frac{|(\mathbf{\Lambda}_t^a - \bar{\mathbf{\Lambda}}_t^a)_{jj}|}{(\bar{\mathbf{\Lambda}}_t^a)_{jj}} = 1 - \frac{1}{\sqrt{1 + \rho_j}} = \frac{\rho_j}{\sqrt{1 + \rho_j}(\sqrt{1 + \rho_j} + 1)} \le \frac{\rho_j}{2}. \tag{22}$$

The assumed bound and the geometric sum yield

$$(\mathbf{d}_{t,a})_j \le C(1 - \beta_2^h)((\mathbf{V}_{t_0-1})_{aj} + \epsilon). \tag{23}$$

Since $b_j \ge \beta_2^h((\mathbf{V}_{t_0-1})_{aj} + \epsilon)$, we obtain

$$\theta_{t,a} \le \frac{1}{2}\max_j \rho_j \le \frac{C}{2}\frac{1 - \beta_2^h}{\beta_2^h}. \tag{24}$$

Finally, $1 - \beta_2^h = (1 - \beta_2)\sum_{k=0}^{h-1}\beta_2^k \le h(1 - \beta_2)$ proves the second inequality. □

For a numerical illustration, the comparison condition in Lemma 9 gives $\rho_j \le C(1 - \beta_2^h)/\beta_2^h$. Since $x \mapsto 1 - (1 + x)^{-1/2}$ is increasing for $x \ge 0$, the exact coordinatewise expression above yields the sharper bound

$$\theta_{t,a} \le 1 - \left(1 + C\frac{1 - \beta_2^h}{\beta_2^h}\right)^{-1/2}. \tag{25}$$

For $h = 200$ and $\beta_2 = 0.999$, we have $\beta_2^h \approx 0.81865$ and $1 - \beta_2^h \approx 0.18135$. The geometric-sum bound in Lemma 9 is then $C(1 - \beta_2^h)/(2\beta_2^h) \approx 0.11076C$. If the comparison condition holds with $C = 1$, equation 25 gives

$$\theta_{t,a} \le 1 - \beta_2^{h/2} = 1 - 0.999^{100} \approx 0.09521. \tag{26}$$

Thus each adaptive scaling coefficient changes by at most approximately 9.52% relative to its reference value. Theorem 3 consequently gives

$$\|\boldsymbol{\mathcal{R}}_{t,a}\|_2 \le (1 - 0.999^{100})\,\kappa(\mathbf{\Omega}_t)\|\mathbf{Z}_{t,a}^0\|_2. \tag{27}$$

For scaled Stiefel projections, $\kappa(\mathbf{\Omega}_t) = 1$, so this also gives a relative remainder bound of approximately 9.52%. For Gaussian projections with $n = 4096$ and $r = 32$, we have $\kappa(\mathbf{\Omega}_t) \le 1.32356$ with probability at least 0.99; on this event, the comparison condition with $C = 1$ yields a relative remainder bound of approximately 12.60%.

B.5 Proof of Proposition 4

Let $N_\tau = K/\tau$ and $N = N_\tau r = Kr/\tau$. Then $\mathbf{C}_K(\tau) = N_\tau^{-1} \sum_{b=1}^{N_\tau} \mathbf{\Omega}^{(b)} (\mathbf{\Omega}^{(b)})^\top$. For Gaussian projections, write $\mathbf{C}_K(\tau) = \mathbf{Z}\mathbf{Z}^\top/N$, where $\mathbf{Z} = \sqrt{r}[\mathbf{\Omega}^{(1)}, \dots, \mathbf{\Omega}^{(N_\tau)}] \in \mathbb{R}^{n \times N}$ has independent standard Gaussian entries. Set $\mathbf{A} = \mathbf{Z}^\top \in \mathbb{R}^{N \times n}$, $u_\delta = \sqrt{2\log(2/\delta)}$, and $\xi_\delta = (\sqrt{n} + u_\delta)/\sqrt{N}$. Applying Vershynin (2012, Corollary 5.35) to $\mathbf{A}$ gives, with probability at least $1 - 2e^{-u_\delta^2/2} = 1 - \delta$,

$$\frac{\sigma_{\max}(\mathbf{A})}{\sqrt{N}} \le 1 + \xi_\delta, \qquad \frac{\sigma_{\min}(\mathbf{A})}{\sqrt{N}} \ge 1 - \xi_\delta \quad \text{when } N \ge n. \tag{28}$$

Since $\mathbf{C}_K(\tau) = \mathbf{A}^\top \mathbf{A}/N$, its eigenvalues are the squared singular values divided by N, with additional zeros when $N < n$. In the latter case $\xi_\delta > 1$, so in both cases

$$\big(\max\{0, 1 - \xi_\delta\}\big)^2 \le \lambda_{\min}(\mathbf{C}_K(\tau)) \le \lambda_{\max}(\mathbf{C}_K(\tau)) \le (1 + \xi_\delta)^2. \tag{29}$$

Therefore,

$$\begin{aligned} \|\mathbf{C}_K(\tau) - \mathbf{I}_n\|_2 &= \max\{\lambda_{\max}(\mathbf{C}_K(\tau)) - 1, 1 - \lambda_{\min}(\mathbf{C}_K(\tau))\} \\ &\le 2\xi_\delta + \xi_\delta^2. \end{aligned} \tag{30}$$

Finally, $\xi_\delta^2 \le 3q_\delta/N$ by Cauchy–Schwarz, and hence

$$2\xi_\delta + \xi_\delta^2 \le 2\sqrt{3q_\delta/N} + 3q_\delta/N \le 4\log 9 \left(\sqrt{q_\delta/N} + q_\delta/N\right). \tag{31}$$

Substituting $N = Kr/\tau$ gives the stated bound.

For scaled Stiefel projections and a fixed unit vector $\mathbf{u}$, let $Y_b = r\|(\mathbf{\Omega}^{(b)})^\top \mathbf{u}\|_2^2$. Rotational invariance gives $Y_b \overset{d}{=} nU$, where $U = \sum_{j=1}^r v_j^2$ and $\mathbf{v}$ is uniform on the unit sphere. If $Z \sim \chi_n^2$ is independent of $\mathbf{v}$, then $ZU \sim \chi_r^2$ and $\mathbb{E}[ZU \mid U] = nU$. Conditional Jensen gives

$$\mathbb{E}e^{\lambda Y_b} \le \mathbb{E}e^{\lambda ZU} = (1 - 2\lambda)^{-r/2}, \qquad \lambda < 1/2. \tag{32}$$

Independence across periods yields the χ_N^2 Chernoff bound for $\sum_b Y_b = N\mathbf{u}^\top \mathbf{C}_K(\tau)\mathbf{u}$. Applying the standard $1/4$-net argument with at most 9^n points (Vershynin, 2012, Lemmas 5.2 and 5.4) gives, with probability at least $1 - \delta$,

$$\|\mathbf{C}_K(\tau) - \mathbf{I}_n\|_2 \le 4\sqrt{s/N} + 4s/N, \qquad s = n\log 9 + \log(2/\delta). \tag{33}$$

Using $s \le (\log 9) q_\delta$ and $N = Kr/\tau$ proves the claim. □

B.6 Proof of Theorem 6

B.6.1 Definitions

We write the full parameter vector as $\boldsymbol{\theta} = (\mathbf{X}^1, \dots, \mathbf{X}^p, \mathbf{X}^{\text{norm}})$, where $\mathbf{X}^l \in \mathbb{R}^{m_l \times n_l}$, $p \ge 1$, and $\mathbf{X}^{\text{norm}} \in \mathbb{R}^d$ with $d \ge 0$. For another element of the same parameter space, write $\boldsymbol{\vartheta} = (\mathbf{Y}^1, \dots, \mathbf{Y}^p, \mathbf{Y}^{\text{norm}})$, with blocks of the same dimensions as those of $\boldsymbol{\theta}$. The full parameter space has the product inner product and norm

$$\begin{aligned} \langle \boldsymbol{\theta}, \boldsymbol{\vartheta} \rangle &= \sum_{l=1}^p \langle \mathbf{X}^l, \mathbf{Y}^l \rangle_F + \langle \mathbf{X}^{\text{norm}}, \mathbf{Y}^{\text{norm}} \rangle, \\ \|\boldsymbol{\theta}\|^2 &= \sum_{l=1}^p \|\mathbf{X}^l\|_F^2 + \|\mathbf{X}^{\text{norm}}\|_2^2. \end{aligned} \tag{34}$$

The uncompressed block may concatenate several vectors; when $d = 0$, its terms are omitted. All components of $\mathbf{G}_t$ are evaluated at the same $\boldsymbol{\theta}_t$, and all parameter blocks are updated simultaneously.

For each compressed block, let

$$\begin{aligned} &\mathbf{U}_t^l = \mathbf{G}_t^l \mathbf{\Omega}_t^l, \qquad \mathbf{V}_t^l = \beta_2 \mathbf{V}_{t-1}^l + (1 - \beta_2)(\mathbf{U}_t^l)^{\odot 2}, \\ &\widehat{\mathbf{V}}_t^l = \frac{\mathbf{V}_t^l}{1 - \beta_2^t} \quad (t \ge 1), \qquad \mathbf{V}_0^l = \widehat{\mathbf{V}}_0^l = 0. \end{aligned} \tag{35}$$

For the uncompressed block, use the same second-moment recursion without a projection:

$$\begin{aligned}\mathbf{v}_t^{\mathrm{norm}} &= \beta_2 \mathbf{v}_{t-1}^{\mathrm{norm}} + (1-\beta_2)(\mathbf{G}_t^{\mathrm{norm}})^{\odot 2},\\ \widehat{\mathbf{v}}_t^{\mathrm{norm}} &= \frac{\mathbf{v}_t^{\mathrm{norm}}}{1-\beta_2^t} \quad (t \geq 1), \qquad \mathbf{v}_0^{\mathrm{norm}} = \widehat{\mathbf{v}}_0^{\mathrm{norm}} = 0.\end{aligned} \tag{36}$$

Define the proposed direction $\mathbf{D}_t$ blockwise by

$$\begin{aligned}\mathbf{D}_t^l &= \left(\frac{\mathbf{G}_t^l \mathbf{\Omega}_t^l}{\sqrt{\widehat{\mathbf{V}}_{t-1}^l + \epsilon}}\right)(\mathbf{\Omega}_t^l)^\top,\\ \mathbf{D}_t^{\mathrm{norm}} &= \frac{\mathbf{G}_t^{\mathrm{norm}}}{\sqrt{\widehat{\mathbf{v}}_{t-1}^{\mathrm{norm}} + \epsilon}}.\end{aligned} \tag{37}$$

The actual direction is $\mathbf{d}_1 = 0$ and $\mathbf{d}_t = \mathbf{D}_t$ for $t \geq 2$, so Algorithm 1 is

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha \mathbf{d}_t. \tag{38}$$

At $t = 1$, only the second-moment states are updated and $\boldsymbol{\theta}_2 = \boldsymbol{\theta}_1$; $\mathbf{D}_1$ is an auxiliary direction for the moment bounds.

We take a deterministic initialization $\boldsymbol{\theta}_1$. Let $\mathcal{F}_{t-1}$ contain all randomness through step $t-1$, so $\boldsymbol{\theta}_t$, $\mathbf{g}_t := \nabla f(\boldsymbol{\theta}_t)$, and the historical states are measurable with respect to $\mathcal{F}_{t-1}$. Write $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot \mid \mathcal{F}_{t-1}]$. Assumption 5 gives

$$\mathbb{E}_t \mathbf{G}_t = \mathbf{g}_t, \qquad \mathbb{E}_t \|\mathbf{G}_t - \mathbf{g}_t\|^2 \leq \sigma^2, \qquad \|\mathbf{g}_t\| \leq R. \tag{39}$$

Given the history and $\mathbf{G}_t$, each current projector retains its independent Gaussian-entry distribution. These conditions concern the current sampling step; they allow the iterates and historical states to depend on all past projections.

For the following bounds, we define

$$\begin{aligned}B &:= \sqrt{R^2+\sigma^2}, \qquad \kappa_l := \frac{n_l + r_l + 1}{r_l}, \qquad \kappa_{\mathrm{G}} := \max_{1\leq l\leq p} \kappa_l, \qquad M := B\sqrt{\kappa_{\mathrm{G}}/\epsilon},\\ W_\beta &:= \frac{\sqrt{1-\beta_2}}{1-\sqrt{\beta_2}}, \qquad J_\beta := \frac{\sqrt{1-\beta_2}}{(1-\sqrt{\beta_2})^2}, \qquad \eta := \frac{1}{BW_\beta + \sqrt{\epsilon}},\\ K_{\mathrm{hist}} &:= 4RBLMJ_\beta, \qquad C_{\mathrm{hist}} := \eta^2 K_{\mathrm{hist}}, \qquad \mathcal{C} := C_{\mathrm{hist}} + \frac{LM^2}{2}.\end{aligned} \tag{40}$$

The main-text considers a common projection rank r across layers, with $n := \max_l n_l$ denoting their maximum input dimension. In this case, $\kappa_{\mathrm{G}} = \max_l (n_l + r + 1)/r = (n + r + 1)/r$, so the constants used here agree with those in the main text. We also write $\Delta_f := f(\boldsymbol{\theta}_1) - f_\star$.

B.6.2 Supporting Lemmas

Lemma 10 (Gaussian back-projection moments) *Let $\mathbf{\Omega} \in \mathbb{R}^{n\times r}$ have independent $\mathcal{N}(0, 1/r)$ entries. For a fixed nonnegative diagonal matrix $\mathbf{\Lambda} \in \mathbb{R}^{r\times r}$ and a fixed row vector $\mathbf{u} \in \mathbb{R}^{1\times n}$,*

$$\begin{aligned}\mathbb{E}[\mathbf{\Omega}\mathbf{\Lambda}\mathbf{\Omega}^\top] &= \frac{\operatorname{tr}\mathbf{\Lambda}}{r}\mathbf{I}_n,\\ \mathbb{E}\|\mathbf{u}\mathbf{\Omega}\mathbf{\Lambda}\mathbf{\Omega}^\top\|_2^2 &= \frac{(\operatorname{tr}\mathbf{\Lambda})^2 + (n+1)\operatorname{tr}(\mathbf{\Lambda}^2)}{r^2}\|\mathbf{u}\|_2^2.\end{aligned} \tag{41}$$

Proof of Lemma 10. Write $\mathbf{\Omega} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_r)$ and $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_r)$. The first identity follows from $\mathbb{E}[\boldsymbol{\omega}_j \boldsymbol{\omega}_j^\top] = \mathbf{I}_n/r$. For the second, put $\mathbf{H} = \mathbf{\Omega}\mathbf{\Lambda}\mathbf{\Omega}^\top$. Independence between columns gives, for $j \neq k$,

$$\mathbb{E}[\boldsymbol{\omega}_j \boldsymbol{\omega}_j^\top \boldsymbol{\omega}_k \boldsymbol{\omega}_k^\top] = \frac{1}{r^2}\mathbf{I}_n. \tag{42}$$

For $j = k$, the diagonal entries of $\mathbb{E}[(\boldsymbol{\omega}_j \boldsymbol{\omega}_j^\top)^2]$ are $(3 + n - 1)/r^2$, and the off-diagonal entries vanish by symmetry. Thus

$$\mathbb{E}\mathbf{H}^2 = \frac{(n+2)\sum_j \lambda_j^2 + \sum_{j\neq k}\lambda_j\lambda_k}{r^2}\mathbf{I}_n = \frac{(\operatorname{tr}\mathbf{\Lambda})^2 + (n+1)\operatorname{tr}(\mathbf{\Lambda}^2)}{r^2}\mathbf{I}_n. \tag{43}$$

Since $\mathbf{H}$ is symmetric, $\|\mathbf{u}\mathbf{H}\|_2^2 = \mathbf{u}\mathbf{H}^2\mathbf{u}^\top$, which proves the claim. □

Lemma 11 (Conditional moment bounds) *Define*

$$P_t^2 := \sum_{l=1}^{p} \|\mathbf{U}_t^l\|_F^2 + \|\mathbf{G}_t^{\mathrm{norm}}\|_2^2. \tag{44}$$

Then, for every $t \geq 1$,

$$\mathbb{E}_t\|\mathbf{G}_t\|^2 \leq B^2, \qquad \mathbb{E}_t P_t^2 \leq B^2, \qquad \mathbb{E}_t P_t \leq B, \qquad \mathbb{E}_t\|\mathbf{D}_t\|^2 \leq M^2. \tag{45}$$

The last bound also holds for the actual direction $\mathbf{d}_t$.

Proof of Lemma 11. Conditional unbiasedness and the variance bound give

$$\mathbb{E}_t\|\mathbf{G}_t\|^2 = \|\mathbf{g}_t\|^2 + \mathbb{E}_t\|\mathbf{G}_t - \mathbf{g}_t\|^2 \leq R^2 + \sigma^2 = B^2. \tag{46}$$

Conditioning additionally on $\mathbf{G}_t$, isotropy gives $\mathbb{E}[\|\mathbf{G}_t^l\mathbf{\Omega}_t^l\|_F^2 \mid \mathcal{F}_{t-1}, \mathbf{G}_t] = \|\mathbf{G}_t^l\|_F^2$. Summing over blocks and applying the tower property proves $\mathbb{E}_t P_t^2 = \mathbb{E}_t\|\mathbf{G}_t\|^2 \leq B^2$; conditional Jensen then gives $\mathbb{E}_t P_t \leq B$.

For row a of block l, define the history-measurable matrix

$$\mathbf{\Lambda}_{t,a}^l := \mathrm{diag}_j\left(((\widehat{\mathbf{V}}_{t-1}^l)_{aj} + \epsilon)^{-1/2}\right). \tag{47}$$

Every diagonal entry is at most $\epsilon^{-1/2}$. Hence

$$\frac{(\mathrm{tr}\,\mathbf{\Lambda}_{t,a}^l)^2 + (n_l+1)\,\mathrm{tr}((\mathbf{\Lambda}_{t,a}^l)^2)}{r_l^2} \leq \frac{r_l^2 + (n_l+1)r_l}{r_l^2\epsilon} = \frac{\kappa_l}{\epsilon}. \tag{48}$$

Apply Lemma 10 conditional on $\mathcal{F}_{t-1}, \mathbf{G}_t$, and then average over $\mathbf{G}_t$:

$$\mathbb{E}_t\|\mathbf{D}_{t,a}^l\|_2^2 \leq \frac{\kappa_l}{\epsilon}\mathbb{E}_t\|\mathbf{G}_{t,a}^l\|_2^2. \tag{49}$$

The uncompressed direction satisfies $\|\mathbf{D}_t^{\mathrm{norm}}\|_2^2 \leq \epsilon^{-1}\|\mathbf{G}_t^{\mathrm{norm}}\|_2^2$. Summing in the product norm yields $\mathbb{E}_t\|\mathbf{D}_t\|^2 \leq \kappa_{\mathrm{G}}B^2/\epsilon = M^2$. Since $\mathbf{d}_t$ is either zero or $\mathbf{D}_t$, the same bound applies to $\mathbf{d}_t$. □

Lemma 12 (Conditional first-order descent) *Let $A_t := \|\mathbf{g}_t\|^2$ and define the total raw and corrected states by*

$$S_t := \sum_{l,a,j}(\mathbf{V}_t^l)_{aj} + \sum_i (\mathbf{v}_t^{\mathrm{norm}})_i, \qquad \widehat{S}_t := \frac{S_t}{1-\beta_2^t}\ (t \geq 1), \qquad \widehat{S}_0 := 0. \tag{50}$$

With

$$\zeta_{t,a}^l := \frac{\mathrm{tr}\,\mathbf{\Lambda}_{t,a}^l}{r_l}, \qquad h_{t,i} := ((\widehat{\mathbf{v}}_{t-1}^{\mathrm{norm}})_i + \epsilon)^{-1/2}, \tag{51}$$

the proposed direction satisfies

$$\mathbb{E}_t\langle \mathbf{g}_t, \mathbf{D}_t\rangle = \sum_{l,a}\zeta_{t,a}^l\|\mathbf{g}_{t,a}^l\|_2^2 + \sum_i h_{t,i}(\mathbf{g}_{t,i}^{\mathrm{norm}})^2 \geq \frac{A_t}{\sqrt{\widehat{S}_{t-1}+\epsilon}}. \tag{52}$$

For $t \geq 2$ the same identity and bound hold for $\mathbf{d}_t$.

Proof of Lemma 12. Fix the history. By conditional independence and the first identity in Lemma 10,

$$\begin{aligned} \mathbb{E}_t\mathbf{D}_{t,a}^l &= \mathbb{E}_t[\mathbf{G}_{t,a}^l\mathbf{\Omega}_t^l\mathbf{\Lambda}_{t,a}^l(\mathbf{\Omega}_t^l)^\top] = \zeta_{t,a}^l\mathbf{g}_{t,a}^l, \\ \mathbb{E}_t\mathbf{D}_{t,i}^{\mathrm{norm}} &= h_{t,i}\mathbf{g}_{t,i}^{\mathrm{norm}}. \end{aligned} \tag{53}$$

Taking the inner product with the history-measurable $\mathbf{g}_t$ proves the equality. Every corrected state entry is nonnegative and bounded by their sum $\widehat{S}_{t-1}$, so all inverse denominators, and their row averages, are at least $(\widehat{S}_{t-1}+\epsilon)^{-1/2}$. This proves the inequality. □

Lemma 13 (Historical mixed moment) *For every $t \geq 1$,*

$$\mathbb{E}[A_t\sqrt{\widehat{S}_{t-1}}] \leq BW_\beta\mathbb{E}A_t + \alpha K_{\mathrm{hist}}. \tag{54}$$

Proof of Lemma 13. We first compare the current gradient with a historical gradient, then sum the resulting bounds using the corrected second-moment weights.

Fix $1 \le s < t$. Smoothness, the bound $\|\mathbf{g}_j\| \le R$, and the actual update equation 38 give

$$\begin{aligned} |A_t - A_s| &\le (\|\mathbf{g}_t\| + \|\mathbf{g}_s\|)\|\mathbf{g}_t - \mathbf{g}_s\| \\ &\le 2RL\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_s\| \le 2RL\alpha \sum_{j=s}^{t-1} \|\mathbf{d}_j\|. \end{aligned} \tag{55}$$

Lemma 11 implies $\mathbb{E}\|\mathbf{d}_j\| \le M$, so

$$\mathbb{E}|A_t - A_s| \le 2RLM\alpha(t-s). \tag{56}$$

Because A_s is $\mathcal{F}_{s-1}$-measurable,

$$\mathbb{E}[P_s A_s] = \mathbb{E}[A_s \mathbb{E}_s P_s] \le B\mathbb{E}A_s. \tag{57}$$

For every $j \ge s$, Cauchy–Schwarz and the unconditional second-moment bounds give

$$\mathbb{E}[P_s\|\mathbf{d}_j\|] \le \sqrt{\mathbb{E}P_s^2}\sqrt{\mathbb{E}\|\mathbf{d}_j\|^2} \le BM. \tag{58}$$

Combining the last three estimates yields

$$\begin{aligned} \mathbb{E}[P_s A_t] &\le \mathbb{E}[P_s A_s] + \mathbb{E}[P_s|A_t - A_s|] \\ &\le B\mathbb{E}A_s + 2RBLM\alpha(t-s) \\ &\le B\mathbb{E}A_t + 4RBLM\alpha(t-s). \end{aligned} \tag{59}$$

The last inequality uses equation 56 once more.

The state recursions imply $S_0 = 0$ and $S_t = \beta_2 S_{t-1} + (1-\beta_2)P_t^2$. For $t \ge 2$, set $m = t-1$ and $\rho = \sqrt{\beta_2} \in (0,1)$. Expanding the second-moment recursion and applying $\|\mathbf{x}\|_2 \le \|\mathbf{x}\|_1$ gives

$$\begin{aligned} \sqrt{\widehat{S}_{t-1}} &= \left(\frac{1-\beta_2}{1-\beta_2^m}\sum_{s=1}^{m}\beta_2^{m-s}P_s^2\right)^{1/2} \le \sum_{s=1}^{m} a_{t,s}P_s, \\ a_{t,s} &:= \sqrt{\frac{1-\beta_2}{1-\beta_2^m}}\rho^{m-s}. \end{aligned} \tag{60}$$

The sum of these weights satisfies

$$\sum_{s=1}^{m} a_{t,s} = \sqrt{\frac{1-\beta_2}{1-\rho^{2m}}}\frac{1-\rho^m}{1-\rho} = W_\beta\sqrt{\frac{1-\rho^m}{1+\rho^m}} \le W_\beta. \tag{61}$$

To control the weighted lag, use the finite-sum identity

$$(1-\rho)\sum_{k=1}^{m} k\rho^{k-1} = \sum_{k=1}^{m}\rho^{k-1} - m\rho^m \le \sum_{k=1}^{m}\rho^{k-1}. \tag{62}$$

It follows that

$$\sum_{s=1}^{m}(t-s)a_{t,s} = \sqrt{\frac{1-\beta_2}{1-\rho^{2m}}}\sum_{k=1}^{m} k\rho^{k-1} \le \frac{W_\beta}{1-\rho} = J_\beta. \tag{63}$$

Both estimates retain the bias-correction factor $1-\beta_2^m$. Multiplying the square-root bound by A_t, taking expectation, and combining equation 59 with the two weight bounds gives

$$\begin{aligned} \mathbb{E}[A_t\sqrt{\widehat{S}_{t-1}}] &\le \sum_{s=1}^{t-1} a_{t,s}\big(B\mathbb{E}A_t + 4RBLM\alpha(t-s)\big) \\ &\le BW_\beta\mathbb{E}A_t + 4RBLM\alpha J_\beta, \end{aligned} \tag{64}$$

This proves the bound for $t \ge 2$. For $t = 1$, $\widehat{S}_0 = 0$, and the result follows directly. □

Lemma 14 (Unconditional first-order descent) *For every $t \ge 2$, the actual direction satisfies*

$$\mathbb{E}\langle \mathbf{g}_t, \mathbf{d}_t\rangle \ge \eta\mathbb{E}\|\mathbf{g}_t\|^2 - \alpha C_{\mathrm{hist}}. \tag{65}$$

Proof of Lemma 14. For $S \geq 0$, $\sqrt{S+\epsilon} \leq \sqrt{S} + \sqrt{\epsilon}$. Define $\phi(y) = (y + \sqrt{\epsilon})^{-1}$ for $y \geq 0$. Since $\phi''(y) = 2(y + \sqrt{\epsilon})^{-3} > 0$, its tangent at $y_0 = BW_\beta$ is a global lower bound:

$$\phi(y) \geq \phi(y_0) + \phi'(y_0)(y - y_0) = \eta - \eta^2(y - BW_\beta). \tag{66}$$

Taking the full expectation in Lemma 12, then applying this tangent bound and Lemma 13, yields

$$\begin{aligned}
\mathbb{E}\langle \mathbf{g}_t, \mathbf{d}_t \rangle &\geq \mathbb{E}\frac{A_t}{\sqrt{\widehat{S}_{t-1} + \epsilon}} \\
&\geq \mathbb{E}[A_t \phi(\sqrt{\widehat{S}_{t-1}})] \\
&\geq \eta \mathbb{E} A_t - \eta^2 \big(\mathbb{E}[A_t \sqrt{\widehat{S}_{t-1}}] - BW_\beta \mathbb{E} A_t\big) \\
&\geq \eta \mathbb{E} A_t - \alpha \eta^2 K_{\text{hist}} = \eta \mathbb{E} A_t - \alpha C_{\text{hist}}.
\end{aligned} \tag{67}$$

This proves the lemma. □

B.6.3 Convergence Proof

For $t \geq 2$, by equation 38, we have

$$f(\boldsymbol{\theta}_{t+1}) \leq f(\boldsymbol{\theta}_t) - \alpha \langle \mathbf{g}_t, \mathbf{d}_t \rangle + \frac{L\alpha^2}{2} \|\mathbf{d}_t\|^2. \tag{68}$$

Taking expectations and applying Lemmas 11 and 14,

$$\mathbb{E} f(\boldsymbol{\theta}_{t+1}) \leq \mathbb{E} f(\boldsymbol{\theta}_t) - \alpha \eta \mathbb{E} \|\mathbf{g}_t\|^2 + \alpha^2 \left(C_{\text{hist}} + \frac{LM^2}{2} \right) = \mathbb{E} f(\boldsymbol{\theta}_t) - \alpha \eta \mathbb{E} A_t + \alpha^2 \mathcal{C}. \tag{69}$$

Sum equation 69 from $t = 2$ to T. Since $\boldsymbol{\theta}_2 = \boldsymbol{\theta}_1$ and $f \geq f_\star$,

$$\alpha \eta \sum_{t=2}^{T} \mathbb{E} \|\mathbf{g}_t\|^2 \leq f(\boldsymbol{\theta}_1) - \mathbb{E} f(\boldsymbol{\theta}_{T+1}) + \alpha^2 (T-1) \mathcal{C} \leq \Delta_f + \alpha^2 (T-1) \mathcal{C}. \tag{70}$$

Including the $t = 1$ term in the average, we obtain

$$\begin{aligned}
\frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \|\mathbf{g}_t\|^2 &\leq \frac{\|\mathbf{g}_1\|^2}{T} + \frac{\Delta_f}{\alpha \eta T} + \frac{\alpha (T-1) \mathcal{C}}{\eta T} \\
&\leq \frac{\|\mathbf{g}_1\|^2}{T} + \frac{1}{\eta T} \left(\frac{\Delta_f}{\alpha} + \alpha T \mathcal{C} \right).
\end{aligned} \tag{71}$$

In particular, a fixed $\alpha = \gamma / \sqrt{T}$, $\gamma > 0$, gives

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \|\mathbf{g}_t\|^2 \leq \frac{\|\mathbf{g}_1\|^2}{T} + \frac{1}{\eta \sqrt{T}} \left(\frac{\Delta_f}{\gamma} + \mathcal{C} \gamma \right). \tag{72}$$

If $B = 0$, all stochastic and true gradients vanish almost surely, so the claim holds for any finite step size. If $\Delta_f = 0 < B$, the prescribed step size is zero and the iterates remain at a global minimizer, so the result is immediate. For $\Delta_f > 0$, $B > 0$, and $L > 0$, we have $\mathcal{C} > 0$. The expression $\Delta_f / \gamma + \mathcal{C} \gamma$ is minimized at $\gamma_\star = \sqrt{\Delta_f / \mathcal{C}}$, giving

$$\alpha_\star = \sqrt{\frac{\Delta_f}{\mathcal{C} T}}, \qquad \frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \|\mathbf{g}_t\|^2 \leq \frac{\|\mathbf{g}_1\|^2}{T} + \frac{2\sqrt{\Delta_f \mathcal{C}}}{\eta \sqrt{T}}. \tag{73}$$

Let $q = 1 - \sqrt{\beta_2}$ and $a = \sqrt{1 - \beta_2}$. To express the optimized bound explicitly, define

$$C_0 := Ba + \sqrt{\epsilon} q, \qquad C_1 := \kappa_{\text{G}} C_0^2 + 8R \sqrt{\epsilon \kappa_{\text{G}}}\, a. \tag{74}$$

Using equation 40, direct substitution gives

$$\eta = \frac{q}{C_0}, \qquad \mathcal{C} = \frac{q^2}{C_0^2}\, 4RBL\left(B\sqrt{\frac{\kappa_{\mathrm{G}}}{\epsilon}}\right)\frac{a}{q^2} + \frac{L\kappa_{\mathrm{G}}B^2}{2\epsilon} = \frac{LB^2C_1}{2\epsilon C_0^2}. \tag{75}$$

Thus the step size in equation 73 is precisely

$$\alpha_\star = \sqrt{\frac{2\epsilon C_0^2 \Delta_f}{LT(R^2+\sigma^2)C_1}}, \tag{76}$$

Substituting yields

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\boldsymbol{\theta}_t)\|^2 \leq \frac{\|\nabla f(\boldsymbol{\theta}_1)\|^2}{T} + \frac{\sqrt{2L(R^2+\sigma^2)C_1\Delta_f}}{\sqrt{\epsilon T}(1-\sqrt{\beta_2})} = \mathcal{O}(T^{-1/2}). \tag{77}$$

This proves Theorem 6. □

C Additional Related Work

Parameter-efficient fine-tuning. Parameter-efficient fine-tuning (PEFT) encompasses a well-established family of methods that reduce gradient and optimizer-state memory by freezing most pretrained weights and optimizing only a small set of parameters. Low-rank adaptation methods parameterize weight updates with compact low-rank factors (Hu et al., 2022; Zhang et al., 2023; Valipour et al., 2023). Additive approaches instead introduce lightweight trainable modules or continuous prompts while keeping the backbone fixed (Houlsby et al., 2019; Lester et al., 2021; Li & Liang, 2021). These methods substantially reduce trainable parameters and optimizer-state memory. However, their restricted update spaces can limit adaptation capacity and lead to worse performance than full-parameter supervised fine-tuning on challenging tasks (Xia et al., 2024).

Optimizer-state compression. Beyond PEFT, optimizer-state quantization is another widely used approach to reducing training memory. Representative methods compress Adam states to 8 or even 4 bits using dedicated quantization schemes (Dettmers et al., 2022; Li et al., 2023). However, quantization errors can impair optimization stability and may lead to worse performance than full-precision full-parameter fine-tuning. Beyond quantization, sparsification-based optimizer-state compression (Modoranu et al., 2024; Zhou et al., 2025) reduces memory by retaining optimizer information only for selected gradient coordinates or parameters rather than maintaining dense moments for the entire model. MicroAdam retains a sliding window of Top-k sparse gradients, reconstructs the Adam statistics from this history, and quantizes the error-feedback buffer, thereby reducing optimizer-state memory while providing convergence guarantees comparable to AMSGrad (Modoranu et al., 2024).

Activation memory reduction. Activations stored for backpropagation require memory that grows approximately linearly with context length, model depth, and hidden dimension, and can dominate peak training memory (Shi et al., 2025). System-level techniques reduce activation memory by avoiding the need to keep all intermediate tensors on the GPU throughout training. Activation checkpointing reduces GPU memory usage by recomputing discarded activations during backpropagation (Chen et al., 2016). Activation offloading instead transfers saved activations to CPU memory until they are needed (Rhu et al., 2016). These memory savings come at the cost of additional recomputation or data-transfer overhead. To mitigate these overheads, recent methods study activation compression in LLMs (Sakr & Khailany, 2024; Liu et al., 2025; Shamshoum et al., 2025; Wei et al., 2026). AGO-CC provides convergence guarantees for unbiased activation compression and introduces an activation–gradient–optimizer co-compression scheme that reuses low-rank activation factors to form compressed linear-layer gradients and maintain low-rank optimizer states. FLAME adopts this co-compression strategy to compute projected gradients directly from compressed activations (Wei et al., 2026).