MEMORY-EFFICIENT SUBSPACE OPTIMIZATION WITH GRADIENT CORRECTION: FROM MOMENTUM SGD TO MUON

Anonymous authors
Paper under double-blind review

ABSTRACT

Memory-efficient subspace optimization has emerged as a practical approach to reducing the memory cost of LLM training while preserving optimization performance. Previous algorithms typically rely on the leading singular subspace of stochastic gradients, yet the gradient noise may dominate the subspace in later training stages and can prevent convergence. To overcome this limitation, we adopt a residual correction framework, which maintains the optimizer states in the subspace and adds the orthogonal residual gradient to the parameter update. We first integrate this scheme into momentum SGD to develop SMRC, which attains an $O(T^{-1/4})$ convergence rate for a fixed orthonormal basis, matching that of full-space momentum SGD. Our analysis introduces a *subspace virtual iterate* to resolve the coupling between the momentum and the residual correction. Furthermore, we extend SMRC to a multi-stage variant which periodically recomputes bases to track the evolving dominant subspace and retains the same rate. Since each basis refresh breaks the consistency of the virtual iterates, we devise a *boundary correction* approach that aligns them across stages. Beyond momentum SGD, we incorporate our framework into Muon and propose PMRC, which applies the matrix sign to the projected momentum. Through a *weighted momentum tracking* analysis that accounts for the residual gradient, we establish an $O(T^{-1/4})$ convergence rate for PMRC, which matches the rate of full-space Muon. We also design a multi-stage PMRC with periodically refreshed bases, which maintains a convergence rate of the same order. Extensive experimental results confirm the effectiveness and efficiency of our algorithms.

1 INTRODUCTION

Large language models (LLMs) (Vaswani et al., 2017; Brown et al., 2020; OpenAI, 2023; Bai et al., 2023; Guo et al., 2025; Team et al., 2025) have demonstrated strong capabilities in dialogue (Thoppilan et al., 2022), language understanding (Chowdhery et al., 2023), mathematical reasoning (Lewkowycz et al., 2022) and code generation (Chen et al., 2021). As model scale grows, training imposes substantial computational and memory demands (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023). The widely used AdamW optimizer (Kingma & Ba, 2015; Loshchilov & Hutter, 2019) further increases memory consumption, since its first- and second-moment estimates are each as large as the model itself. To alleviate the memory overhead, recent work has explored different forms of memory-efficient training, ranging from restricting the set of trainable parameters (Houlsby et al., 2019; Ben Zaken et al., 2022; Hu et al., 2022; Zhang et al., 2023; Pan et al., 2024) to maintaining optimizer states in low-dimensional subspaces (Zhao et al., 2024; He et al., 2025; Chen et al., 2025a; Pan et al., 2025a). In this work, we focus on memory-efficient subspace optimization, which reduces optimization memory by maintaining the optimizer states in a low-dimensional subspace. GaLore (Zhao et al., 2024) is a representative method in this line of work. For a weight matrix $W_t \in \mathbb{R}^{m\times n}$, GaLore utilizes an orthonormal basis $P \in \mathbb{R}^{m\times r}$ with $r \ll \min\{m, n\}$ and projects the gradient $G_t \in \mathbb{R}^{m\times n}$ into the space $\mathbb{R}^{r\times n}$, where the optimizer maintains its states, reducing the memory complexity from $O(mn)$ to $O((m+n)r)$. To construct the projection matrix, GaLore computes the leading singular vectors of the stochastic gradient to capture its dominant directions and periodically recomputes P via SVD during training.

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.

As shown by He et al. (2025), GaLore may fail to converge to a stationary point under standard stochastic assumptions. Since the projection basis is estimated from stochastic gradients, whose leading singular directions can be dominated by anisotropic noise *rather than the true gradient* in later training stages, the selected subspace fails to capture gradient directions. While GoLore (He et al., 2025) replaces the SVD basis with random projections and establishes convergence under standard assumptions, its random subspaces do not exploit dominant gradient directions and are less effective in practice. Recent methods that retain gradient-dependent subspaces either require stronger assumptions (Robert et al., 2025) or offer no memory advantage in the worst case (Pan et al., 2025a). Therefore, there still *remains a lack of convergence guarantees for memory-efficient subspace optimization with gradient-dependent dominant subspaces under standard assumptions.*

Subspace Momentum SGD with Residual Correction (SMRC). Inspired by Chen et al. (2025a), we adopt a residual correction framework, which maintains optimizer states in a low-dimensional subspace and adds the residual gradient in its *orthogonal complement* to the update. We incorporate this framework into momentum SGD (Liu et al., 2020) and propose SMRC, which keeps momentum only in the projected space and reduces optimizer-state memory from $O(mn)$ to $O((m+n)r)$. Although the projected momentum update and the residual correction lie in orthogonal subspaces, they remain coupled across iterations. To overcome this difficulty, we devise a *subspace virtual iterate* strategy that absorbs the momentum into an auxiliary trajectory. SMRC achieves an $O(T^{-1/4})$ convergence rate for any fixed orthonormal basis, matching that of full-space momentum SGD. Furthermore, since the dominant gradient subspace evolves during training, practical methods periodically refresh the projection matrix P, and we accordingly extend SMRC to a multi-stage variant. A direct application of the single-stage analysis results in a mismatch between virtual iterates at stage boundaries. We design a *boundary correction* approach that preserves the consistency of the virtual sequence. Our multi-stage SMRC attains the same rate as that of single-stage SMRC.

Projected Muon with Residual Correction (PMRC). Beyond momentum SGD, we apply the residual correction framework to Muon (Jordan et al., 2024), an effective optimizer for LLM training (Liu et al., 2025; Team et al., 2025), and develop PMRC. It maintains momentum in the projected coordinates and applies the matrix sign to the low-dimensional momentum. This nonlinear dependence of the update on the momentum invalidates the subspace virtual-iterate construction used for SMRC. To address this challenge, we present a *weighted momentum tracking* analysis that jointly characterizes momentum accumulation and effects induced by the residual correction. The matrix-sign step corresponds to steepest descent under the spectral norm and is characterized by the nuclear norm (Bernstein & Newhouse, 2024), whereas the residual gradient correction is measured in the Frobenius norm. We establish convergence under a mixed stationarity measure and PMRC attains the same $O(T^{-1/4})$ rate as that of full-space Muon (Li & Hong, 2025). Similarly, we generalize PMRC to a multi-stage setting with periodically refreshed projection bases that better capture the evolving dominant gradient subspace and improve practical performance. The multi-stage variant retains the same $O(T^{-1/4})$ convergence rate.

Finally, we evaluate our methods through extensive experiments. On synthetic problems, we examine the residual correction and the multi-stage basis refresh in SMRC, and the results show that the residual correction restores convergence while periodic refreshes track the evolving dominant gradient subspace. For practical LLM training, we develop a practical implementation of PMRC and evaluate it on LLaMA pretraining across **four** model scales from 60M to 1.3B parameters and on **six** supervised finetuning settings. Our PMRC consistently performs favorably against prior memory-efficient methods across different model scales and training settings.

2 Related Work

In this section, we briefly review related work on memory-efficient optimization and on Muon and its variants. Due to space limitations, we provide additional discussion in Appendix A.

2.1 Memory-efficient Optimization for LLM Training

As model scale grows, full-parameter training becomes increasingly memory-intensive, since gradients and optimizer states impose substantial overhead. Parameter-efficient finetuning methods reduce this cost by restricting the update to a small set of parameters. LoRA (Hu et al., 2022)

freezes the model weights and learns low-rank updates, which is effective for finetuning but limits the expressiveness of the update in pretraining (Ding et al., 2023; Lialin et al., 2024). GaLore (Zhao et al., 2024) instead reduces memory by maintaining optimizer states in a low-dimensional gradient subspace. Given a stochastic gradient $G_t \in \mathbb{R}^{m\times n}$, where we assume $m \leq n$ without loss of generality, GaLore uses an orthonormal basis $P_t \in \mathbb{R}^{m\times r}$ with $r \ll m$ to form the projected gradient $P_t^\top G_t \in \mathbb{R}^{r\times n}$. The optimizer states are maintained in the reduced coordinates and the resulting update is mapped back to the original parameter space through P_t. GaLore constructs P_t from the top-r left singular vectors of the stochastic gradient. Since computing the top-r SVD at every iteration is expensive, GaLore refreshes P_t once every τ iterations while keeping it fixed between consecutive refreshes. As shown by He et al. (2025), the leading singular subspaces computed from stochastic gradients can be dominated by anisotropic noise rather than the true gradient, leaving informative gradient directions outside $\mathrm{span}(P_t)$ and preventing convergence under standard stochastic assumptions.

To address this limitation, GoLore (He et al., 2025) replaces the SVD basis with random projections. While GoLore establishes convergence guarantees under standard stochastic assumptions, its random subspaces do not preferentially capture dominant gradient directions and lead to slow convergence. Fira (Chen et al., 2025a) follows a complementary approach by retaining the SVD-based subspace and augmenting the projected update with a norm-scaled correction of the discarded residual gradient. Although Fira performs well empirically, its residual scaling is primarily motivated by empirical observations and lacks a convergence guarantee. Zmushko et al. (2025) and Nguyen & Nguyen (2025) also maintain optimizer states in a subspace and apply an SGD-type update in the orthogonal complement. The analysis in Zmushko et al. (2025) is developed for momentum SGD on selected coordinates combined with SGD on the remaining ones. Its convergence guarantee therefore does not directly cover the periodically refreshed SVD subspaces used in GaLore. The convergence rate of Nguyen & Nguyen (2025) is established for a fixed basis under stronger assumptions and applies only to an unweighted residual update.

2.2 Muon and its Variants

Muon (Jordan et al., 2024) has demonstrated strong empirical performance in LLM training, outperforming AdamW while exhibiting stable scaling behavior (Team et al., 2025; Liu et al., 2025). For matrix-shaped parameters, Muon updates along the matrix sign of the momentum, the steepest-descent direction under the spectral norm, and approximates it efficiently through Newton–Schulz iterations. Despite these advantages, Muon still requires a momentum of the same size as the parameter matrix and incurs the same $O(mn)$ optimizer-state memory cost as momentum SGD.

In recent years, a growing body of work has developed Muon variants to improve its memory efficiency (Gupta et al., 2025; Refael et al., 2025; Pan et al., 2025a; Huang et al., 2026; Anson et al., 2026), and we provide a detailed discussion of these methods in Appendix A.3. Among them, GUM (Pan et al., 2025a) combines GaLore-style low-rank projection with layerwise sampling. At each sampling period, a subset of layers receives compensated full-rank updates, while the remaining layers receive projected low-rank ones. The two branches are reweighted according to their sampling probabilities. Since the full-rank branches are drawn independently across layers, all layers may be selected at the same time, in which case GUM offers no memory reduction over Muon.

3 Notation and Assumptions

In this section, we present necessary preliminaries for our work.

Notation. We denote the optimization variable by $W \in \mathbb{R}^{m\times n}$ and assume $m \leq n$ without loss of generality. We use I_r to denote the $r \times r$ identity matrix. For matrices $A, B \in \mathbb{R}^{m\times n}$, we define the trace inner product as $\langle A, B\rangle = \mathrm{tr}(A^\top B)$ and denote the Frobenius, spectral and nuclear norms by $\|A\|_F$, $\|A\|_2$ and $\|A\|_*$, respectively. We refer to a matrix $P \in \mathbb{R}^{m\times r}$ with $P^\top P = I_r$ as an orthonormal basis. The matrix $PP^\top$ is the orthogonal projection matrix onto $\mathrm{span}(P)$ and $Q = I - PP^\top$ is the projection matrix onto $\mathrm{span}(P)^\perp$. All expectations are taken with respect to the randomness of the stochastic gradient oracle.

In the following, we introduce some standard assumptions in stochastic nonconvex optimization (He et al., 2025).

Algorithm 1 Single-Stage SMRC

Input: initial iterate W_1, $P \in \mathbb{R}^{m\times r}$, η, $\beta \in (0,1)$, $\gamma \in (0,1]$

1: Initialize momentum: $\mathbf{m}_0 = \mathbf{0}$
2: **for** $t = 1, \ldots, T$ **do**
3: Query the stochastic gradient $G_t = g(W_t)$
4: Compute $\mathbf{m}_t = (1-\beta)\mathbf{m}_{t-1} + \beta P^\top G_t$
5: Update $W_{t+1} = W_t - \eta\big(P\mathbf{m}_t + \gamma(I - PP^\top)G_t\big)$
6: **end for**

Assumption 1 (Smoothness) The loss function $F(\cdot)$ is L-smooth, i.e., $\|\nabla F(W) - \nabla F(W')\|_F \le L\|W - W'\|_F, \quad \forall W, W' \in \mathbb{R}^{m\times n}$.

Assumption 2 (Lower boundedness) The loss function $F(\cdot)$ is lower bounded, i.e., $F_* = \inf_{W\in\mathbb{R}^{m\times n}} F(W) > -\infty$. We define $\Delta = F(W_1) - F_*$ as the initial optimality gap.

Assumption 3 (Stochastic gradients) For any W_t, the stochastic gradient oracle returns $g(W_t)$ such that $\mathbb{E}\left[g(W_t)\right] = \nabla F(W_t)$ and $\mathbb{E}\left[\|g(W_t) - \nabla F(W_t)\|_F^2\right] \le \sigma^2$.

Assumption 4 (Orthonormal basis) The basis $P_k \in \mathbb{R}^{m\times r}$ is determined by the optimization history prior to stage k and satisfies $P_k^\top P_k = I_r$.

4 Subspace Momentum SGD with Residual Correction (SMRC)

In this section, we incorporate the residual correction framework into momentum SGD and develop SMRC. We first present the single-stage SMRC with a fixed orthonormal basis and establish its convergence guarantee. Building on this result, we then propose a multi-stage variant with periodically refreshed bases and show that it retains the same convergence rate.

4.1 Single-Stage SMRC

As discussed above, GaLore (Zhao et al., 2024) discards the gradient component outside the subspace, which can prevent convergence under stochastic noise. GoLore (He et al., 2025) restores the convergence guarantee through random projections, yet its gradient-independent projection strategy leads to worse practical performance than GaLore. To resolve this dilemma, we adopt a residual correction framework inspired by Chen et al. (2025a), which maintains the optimizer states in the low-rank subspace and adds the orthogonal residual gradient to the parameter update.

In this section, we incorporate the residual correction framework into momentum SGD, a widely used method, and develop SMRC, as summarized in Algorithm 1. At each iteration t, SMRC queries a stochastic gradient G_t, projects it onto the subspace and maintains the low-dimensional momentum $\mathbf{m}_t = (1-\beta)\mathbf{m}_{t-1} + \beta P^\top G_t \in \mathbb{R}^{r\times n}$, where $P \in \mathbb{R}^{m\times r}$ is a fixed orthonormal basis and $\beta \in (0,1)$ is the momentum parameter. When P consists of the top-r left singular vectors of the stochastic gradient, the projection $P^\top G_t$ retains the components along the estimated dominant directions, and the momentum is mapped back to the original parameter space through $P\mathbf{m}_t$. However, as pointed out by He et al. (2025), this subspace may primarily reflect the noise and miss informative gradient directions once the stochastic noise dominates the true gradient. To recover these directions, we incorporate the residual gradient into the update, i.e., $W_{t+1} = W_t - \eta(P\mathbf{m}_t + \gamma(I - PP^\top)G_t)$, where the parameter $\gamma \in (0,1]$ controls the weight of the residual gradient.

We next outline the key technical difficulty introduced by the residual correction, deferring the detailed argument to Appendix D. Since the two update directions are orthogonal, a natural approach is to analyze the projected and residual terms separately. However, this direct decomposition is insufficient, as the residual correction directly affects the iterate, which alters subsequent gradients that are projected and accumulated in the momentum, leaving the two components coupled across iterations. To overcome this challenge, we develop a subspace virtual iterate approach, which introduces the auxiliary sequence $U_t = W_t - \frac{(1-\beta)\eta}{\beta} P\mathbf{m}_{t-1}$ and $U_1 = W_1$. This sequence is used only in the analysis and is not computed by SMRC. The virtual iterate incorporates the projected momentum

Algorithm 2 Multi-Stage SMRC

Input: initial iterate $W_{1,1}$, P_1, η, $\beta \in (0,1)$, $\gamma \in (0,1]$, K, $\{T_k\}_{k=1}^K$

for $k = 1, \dots, K$ **do**
  Set momentum $\mathbf{m}_{k,0} = \mathbf{0}$.
  for $t = 1, \dots, T_k$ **do**
    Query the stochastic gradient $G_{k,t} = g(W_{k,t})$
    Calculate $\mathbf{m}_{k,t} = (1-\beta)\mathbf{m}_{k,t-1} + \beta P_k^\top G_{k,t}$
    Update $W_{k,t+1} = W_{k,t} - \eta\big(P_k \mathbf{m}_{k,t} + \gamma(I - P_k P_k^\top) G_{k,t}\big)$
  end for
  if $k < K$ **then**
    Set $W_{k+1,1} = W_{k,T_k+1} - \frac{(1-\beta)\eta}{\beta} P_k \mathbf{m}_{k,T_k}$ and update P_{k+1}
  end if
end for

into an auxiliary trajectory, thereby decoupling it from the residual correction term. Formally, we establish the theoretical guarantee for SMRC.

Theorem 1 (Single-stage SMRC) *Define $Q = I - PP^\top$ and $c_\beta = \frac{1-\beta}{\beta}$. Under Assumptions 1, 2 and 3, for a matrix $P \in \mathbb{R}^{m\times r}$ with $P^\top P = I_r$ and by setting $\eta = \min\left\{\frac{\gamma}{2L}, \frac{\gamma}{2Lc_\beta}, \frac{1}{L\sqrt{T}}\right\}$, $\beta \in (0,1), \gamma \in (0,1]$, Algorithm 1 ensures*

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[\left\|\nabla F(W_t)\right\|_F\right] = O(T^{-1/4}).$$

Remark 1 For any $\beta \in (0,1), \gamma \in (0,1]$ and the step size prescribed in Theorem 1, SMRC attains an $O(T^{-1/4})$ convergence rate, which is standard for stochastic nonconvex optimization and matches that of full-space momentum SGD, while storing only a low-rank momentum.

Remark 2 SMRC maintains the momentum $\mathbf{m}_t \in \mathbb{R}^{r\times n}$ in the projected space rather than in the ambient space $\mathbb{R}^{m\times n}$. The residual correction $(I - PP^\top)G_t$ can be computed directly from the stochastic gradient and needs no additional optimizer state (Chen et al., 2025a). Storing the basis $P \in \mathbb{R}^{m\times r}$ and the momentum $\mathbf{m}_t$ only requires $O((m+n)r)$ memory, which is smaller than the $O(mn)$ memory of full-space momentum SGD.

4.2 Multi-stage SMRC

Although the single-stage analysis accommodates any fixed orthonormal basis P, practical subspace methods (Zhao et al., 2024; Pan et al., 2025a) typically construct P from the dominant singular subspace of the stochastic gradient so that the projected update retains informative gradient directions. Since the dominant singular subspace of the gradient varies throughout optimization, practical subspace methods periodically update the projection basis to capture the evolving geometry of the loss landscape. Following the periodic refresh strategy of GaLore (Zhao et al., 2024), we propose the multi-stage SMRC as presented in Algorithm 2. It partitions the total N iterations into K stages, where stage k contains T_k iterations and $N = \sum_{k=1}^K T_k$. Within each stage k, SMRC performs T_k updates using a fixed orthonormal basis P_k and maintains the corresponding low-dimensional momentum $\mathbf{m}_{k,t}$. At the end of stage k, a new basis P_{k+1} is computed from the dominant subspace of the stochastic gradient and the subsequent stage proceeds in the refreshed subspace.

The extension of the single-stage analysis to periodically refreshed bases introduces an additional challenge at stage boundaries. If each new stage is initialized with the terminal iterate of the preceding stage, the corresponding virtual sequence becomes *inconsistent* across basis refreshes. To resolve this mismatch, we introduce a *boundary correction* approach that modifies the iterate at each stage transition to compensate for the offset. The corrected transition ensures the terminal virtual iterate of the stage coincides with the initial virtual iterate of the next. The corresponding convergence guarantee of multi-stage SMRC is stated below.

Algorithm 3 Single-Stage PMRC

Input: initial iterate W_1, $P \in \mathbb{R}^{m\times r}$, η, γ, $\beta \in (0,1)$

1: Initialize momentum: $M_0 = \mathbf{0}$
2: **for** $t = 1, \dots, T$ **do**
3: Query the stochastic gradient $G_t = g(W_t)$
4: Calculate $M_t = (1-\beta)M_{t-1} + \beta P^\top G_t$ and $Z_t = \mathrm{msign}(M_t)$
5: Update $W_{t+1} = W_t - \eta\big(PZ_t + \gamma(I - PP^\top)G_t\big)$
6: **end for**

Theorem 2 (Multi-stage SMRC) *Let the total number of iterations be $N = \sum_{k=1}^{K} T_k$, where stage k contains T_k iterations. Define $Q_k = I - P_k P_k^\top$ and $c_\beta = (1-\beta)/\beta$. Under Assumptions 1, 2, 3 and 4, for any orthonormal stage basis P_k with $P_k^\top P_k = I_r$, by setting $\eta = \min\left\{\frac{\gamma}{2L}, \frac{\gamma}{2Lc_\beta}, \frac{1}{L\sqrt{N}}\right\}, \beta \in (0,1), \gamma \in (0,1]$, Algorithm 2 ensures*

$$\frac{1}{N}\sum_{k=1}^{K}\sum_{t=1}^{T_k}\mathbb{E}\left[\left\|\nabla F(W_{k,t})\right\|_F\right] = O(N^{-1/4}).$$

Remark 3 We have established an $O(N^{-1/4})$ convergence rate for multi-stage SMRC over N iterations, which matches the convergence order of full-space momentum SGD (Liu et al., 2020). Notably, Theorem 2 holds for any sequence of orthonormal bases $\{P_k\}_{k=1}^K$. In particular, each P_k may be constructed from the dominant singular subspace of the stochastic gradient.

5 Projected Muon with Residual Correction (PMRC)

In this section, we extend the residual correction framework to Muon (Jordan et al., 2024), a modern optimizer for LLM training. We first present the single-stage method and establish its convergence guarantee and then extend the method and the guarantee to the multi-stage variant.

5.1 Single-Stage PMRC

Muon (Jordan et al., 2024) has demonstrated strong performance in LLM training (Liu et al., 2025; Team et al., 2025), while its full-dimensional momentum incurs the same $O(mn)$ optimizer-state memory cost as momentum SGD. At larger model scales, this memory requirement becomes increasingly restrictive and motivates the development of memory-efficient variants of Muon (Pan et al., 2025a). A natural question is *whether the momentum of Muon can be maintained in a low-dimensional subspace while preserving the convergence guarantee of full-space Muon*.

We incorporate the residual correction framework into Muon and develop PMRC, as outlined in Algorithm 3. Specifically, PMRC projects the stochastic gradient G_t onto the subspace, maintains the low-rank momentum $M_t = (1-\beta)M_{t-1} + \beta P^\top G_t \in \mathbb{R}^{r\times n}$ and applies the matrix sign to M_t. Then, PMRC incorporates the orthogonal residual gradient into the parameter update, that is, $W_{t+1} = W_t - \eta(P\,\mathrm{msign}(M_t) + \gamma(I - PP^\top)G_t)$, where $\gamma > 0$ controls the residual correction.

Notably, PMRC combines the Muon update with residual correction and the two update terms are characterized by *different norm geometries.* The matrix-sign step corresponds to steepest descent under the spectral norm and is characterized by the nuclear norm (Bernstein & Newhouse, 2024; Li & Hong, 2025), whereas the residual gradient correction is measured in the Frobenius norm. We analyze PMRC through a *weighted momentum-tracking* argument for the bias-corrected projected momentum, which accounts for both momentum accumulation and the iterate variation induced by residual correction. We establish convergence under the following mixed stationarity measure.

Theorem 3 (Single-stage PMRC) *Define $Q = I - PP^\top$. Under Assumptions 1, 2 and 3, for a matrix $P \in \mathbb{R}^{m\times r}$ with $P^\top P = I_r$ and by setting $\beta = \Theta(T^{-1/2}), \eta = \Theta(T^{-3/4}), \gamma = \Theta(T^{1/4})$ and $L\eta\gamma \le 1$, Algorithm 3 ensures*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\Big[\left\|P^\top\nabla F(W_t)\right\|_* + \left\|Q\nabla F(W_t)\right\|_F\Big] \le O(T^{-1/4}).$$

Algorithm 4 Multi-Stage PMRC

Input: initial iterate $W_{1,1}$, P_1, η, γ, K, T, $\beta \in (0,1)$

for $k = 1, \dots, K$ **do**
  Set momentum $M_{k,0} = \mathbf{0}$
  for $t = 1, \dots, T$ **do**
    Query the stochastic gradient $G_{k,t} = g(W_{k,t})$
    Calculate $M_{k,t} = (1-\beta)M_{k,t-1} + \beta P_k^\top G_{k,t}$ and $Z_{k,t} = \mathrm{msign}(M_{k,t})$
    Update $W_{k,t+1} = W_{k,t} - \eta\big(P_k Z_{k,t} + \gamma(I - P_k P_k^\top)G_{k,t}\big)$
  end for
  if $k < K$ **then**
    Set $W_{k+1,1} = W_{k,T+1}$ and update P_{k+1}
  end if
end for

Remark 4 Under standard stochastic assumptions, PMRC attains an $O(T^{-1/4})$ convergence rate under the mixed stationarity measure, matching the convergence order of full-space Muon (Li & Hong, 2025). We establish convergence for a GaLore-style Muon variant that maintains its optimizer states entirely in a low-dimensional space.

Remark 5 When $r = m$ and $P = I$, the mixed stationarity measure reduces to $\|\nabla F(W_t)\|_*$, which coincides with the standard stationarity measure for Muon. In this case, PMRC reduces to Muon and recovers the $O(T^{-1/4})$ convergence order of full-space Muon. Moreover, the mixed stationarity measure also upper bounds the Frobenius norm of the full gradient, i.e., $\|\nabla F(W_t)\|_F \le \|P^\top \nabla F(W_t)\|_* + \|(I - PP^\top)\nabla F(W_t)\|_F$. Thus, convergence under the mixed stationarity measure also guarantees convergence of the full gradient in the Frobenius norm.

Remark 6 One may notice that the prescribed choice of γ differs between SMRC and PMRC. Although γ controls the weight of the residual correction relative to the projected update in both methods, the two algorithms are analyzed under different optimization geometries. In SMRC, both the projected and residual components are measured in the Frobenius norm, so $\gamma \in (0, 1]$ is sufficient for the convergence guarantee. In PMRC, the projected Muon component is measured in the nuclear norm, whereas the residual component is measured in the Frobenius norm. The choice $\gamma = \Theta(T^{1/4})$ corresponds to an effective step size $\eta\gamma = \Theta(T^{-1/2})$ for the residual component, the standard step size for SGD, while the Muon component retains $\eta = \Theta(T^{-3/4})$.

5.2 Multi-Stage PMRC

Since the dominant gradient subspace may evolve during optimization, as discussed in Section 4.2, we propose a multi-stage variant of PMRC, as presented in Algorithm 4. At stage k, PMRC constructs an orthonormal basis P_k, initializes the momentum as $M_{k,0} = \mathbf{0}$, and performs T updates with the basis P_k fixed. At the end of stage k, the terminal iterate is used to initialize the subsequent stage, i.e., $W_{k+1,1} = W_{k,T+1}$. The projection basis is then refreshed to P_{k+1}, and the momentum is reinitialized in the corresponding projected coordinates. In this way, each stage retains the single-stage PMRC update with a fixed basis, while the projection subspace is periodically updated to track the evolving dominant gradient directions. We establish the convergence guarantee below.

Theorem 4 (Multi-stage PMRC) *Define $Q_k = I - P_k P_k^\top$. Under Assumptions 1, 2, 3 and 4, with stage lengths T, stage count $K = T$, total iterations $N = KT$ and by setting $\beta = \Theta(N^{-1/2})$, $\eta = \Theta(N^{-3/4})$, $\gamma = \Theta(N^{1/4})$ and $L\eta\gamma \le 1$, Algorithm 4 ensures*

$$\frac{1}{N}\sum_{k=1}^{K}\sum_{t=1}^{T}\mathbb{E}\Big[\big\|P_k^\top \nabla F(W_{k,t})\big\|_* + \big\|Q_k \nabla F(W_{k,t})\big\|_F\Big] \le O(N^{-1/4}).$$

Remark 7 We establish an $O(N^{-1/4})$ convergence rate under the mixed stationarity measure, where N is the total number of stochastic gradient evaluations. This rate is consistent with the single-stage guarantee in Theorem 3 and the standard rate for full-space Muon (Li & Hong, 2025).

Table 1: Training and validation losses during pretraining. **Bold** and underlined values indicate the best and second-best results among memory-efficient methods, respectively.

Loss Type	Memory-Efficient	Optimizer	60M	150M	350M	1.3B
Training Loss	×	AdamW	3.445	3.156	2.843	2.551
	×	Muon	3.383	3.086	2.703	2.414
	✓	GaLore	3.484	3.180	2.972	2.842
	✓	GoLore	3.516	3.196	3.003	3.080
	✓	Fira	3.601	3.180	2.852	2.657
	✓	GUM	3.461	3.141	2.871	2.609
	✓	**PMRC (Ours)**	**3.406**	**3.117**	**2.769**	**2.500**
Validation Loss	×	AdamW	3.421	3.134	2.892	2.613
	×	Muon	3.352	3.067	2.746	2.495
	✓	GaLore	3.462	3.202	2.980	2.875
	✓	GoLore	3.494	3.204	3.015	3.110
	✓	Fira	3.576	3.161	2.909	2.730
	✓	GUM	3.437	3.156	2.881	2.656
	✓	**PMRC (Ours)**	**3.380**	**3.095**	**2.786**	**2.570**

6 Empirical Study

In this section, we assess the performance of the proposed methods through LLM pretraining and finetuning tasks. In experiments, we adopt a practical PMRC variant that applies a residual cap to improve training stability and refreshes the basis through an efficient approximation of SVD, as detailed in Appendix E.2.1. Due to space limitations, we present only a subset of the results and defer the synthetic experiments and ablation studies to Appendix E.1 and Appendix E.2.4, respectively.

6.1 LLM pretraining

We adopt the LLaMA architecture (Touvron et al., 2023) and train models with 60M, 150M, 350M and 1.3B parameters. Their architectural configurations are summarized in Table 4. All models are pretrained on the C4 dataset (Raffel et al., 2020).

Setup. We train the 60M and 150M models on 2 NVIDIA RTX PRO 6000 GPUs, the 350M model on 4 NVIDIA H20 GPUs, and the 1.3B model on 8 NVIDIA H20 GPUs. The corresponding per-device batch sizes are 256, 256, 160, and 160, yielding global batch sizes of 512, 512, 640, and 1280, respectively. The sequence length is 512 for the 60M and 150M models and 1024 for the 350M and 1.3B models. For each model, the number of training steps approximately follows the Chinchilla scaling law (Hoffmann et al., 2022). For example, the 1.3B model is trained for 20480 steps, which corresponds to $20480 \times 1024 \times 1280 \approx 26.84$ billion training tokens, or approximately 20.6 tokens per parameter. We compare our multi-stage PMRC with AdamW (Loshchilov & Hutter, 2019), Muon (Jordan et al., 2024), the AdamW-based GaLore (Zhao et al., 2024), GoLore (He et al., 2025) and Fira (Chen et al., 2025a), and the Muon-based GUM (Pan et al., 2025a). For projection-based methods, we set the projection rank to $r = 256$ and the refresh interval to 100. Since GUM randomly selects a subset of layers for full-rank updates, its memory cost depends on the number of full-rank layers s. For a fair comparison, we choose s according to the memory budget of PMRC, setting $s = 6$ and $r = 128$ for the 60M, 150M, and 350M models, and $s = 4$ and $r = 128$ for the 1.3B model. The peak allocated memory for the 1.3B model is reported in Table 6. The detailed setup, parameter settings and additional pretraining results are deferred to Appendix E.2.2.

Results. Table 1 presents a comparison of the training and validation losses achieved by PMRC with those achieved by AdamW, Muon, GaLore, GoLore, Fira and GUM. As evidenced by the reported results, PMRC consistently achieves the lowest training and validation losses among memory-efficient methods. The improvement supports the effectiveness of residual correction in practical LLM pretraining. PMRC also requires less optimizer-state memory than AdamW while achieving lower losses at every scale. We further evaluate the pretrained models on seven downstream benchmarks, with complete results reported in Appendix E.2.2, where PMRC achieves the highest average accuracy among memory-efficient methods.

Table 2: Accuracy (%) on IFEval and GSM8K benchmarks. **Bold** and underlined values indicate the best and second-best results among memory-efficient methods, respectively.

	IFEval			GSM8K		
Optimizer	**Qwen3-8B**	**LLaMA-3.1-8B**	**OLMo-2-1124-7B**	**Qwen3-8B**	**LLaMA-3.1-8B**	**OLMo-2-1124-7B**
AdamW	65.83	50.12	38.97	90.14	72.10	77.63
Muon	62.11	46.16	36.45	89.76	72.78	78.01
GaLore	66.67	47.24	<u>38.13</u>	89.16	<u>71.27</u>	77.86
GoLore	<u>68.35</u>	<u>47.96</u>	35.73	88.93	70.66	77.10
GUM	62.47	43.65	36.21	<u>89.61</u>	69.74	<u>78.32</u>
PMRC (Ours)	**69.06**	**50.24**	**38.97**	**90.30**	**72.48**	**78.62**

6.2 LLM SUPERVISED FINETUNING

In this section, we conduct supervised finetuning (SFT) of LLMs to enhance their instruction-following and mathematical reasoning abilities. Following previous work (Pan et al., 2025a;b; Yu et al., 2026a), we finetune Qwen3-8B (Yang et al., 2025), LLaMA3.1-8B (Grattafiori et al., 2024) and OLMo-2-1124-7B (OLMo et al., 2024) on two common tasks. For a fair comparison, we set the projection rank to $r = 512$ for all projection-based methods except GUM, for which we use $r = 128$. For GUM, we adjust the number of selected full-rank layers according to the memory cost of PMRC, using $s = 6$, $s = 5$, and $s = 5$ layers for Qwen3-8B, LLaMA3.1-8B, and OLMo-2-1124-7B, respectively. Under these settings, the peak memory usage of GUM still exceeds that of PMRC. We refresh the projection basis every 100 optimizer steps for all memory-efficient optimizers. Our experiments are conducted on a single NVIDIA RTX PRO 6000 GPU and 4 NVIDIA H20 GPUs.

Setup. For the instruction-following task, we follow Pan et al. (2025a) and train models on the GPT-4-LLM dataset, which contains approximately 52K instruction-response examples synthesized by GPT-4. We report performance on IFEval (Zhou et al., 2023), a benchmark designed to measure how reliably a model satisfies explicit instructions and formatting constraints. For the mathematical reasoning task, we follow Yu et al. (2026a) and train models on the NuminaMath-CoT dataset (Li et al., 2024), which consists of 860K problems from high school mathematics and international Olympiad competitions, and we randomly select 64K of them for finetuning. We evaluate the models on the GSM8K benchmark (Cobbe et al., 2021), which consists of grade school math word problems. For both tasks, we use a global batch size of 128 and a maximum sequence length of 2048, and train for one epoch. For evaluation, we allow up to 1280 and 2048 new tokens for IFEval and GSM8K, respectively. We provide detailed training and evaluation configurations, along with peak allocated memory results, in Appendix E.2.3.

Results. We report the instruction-level strict accuracy on the IFEval benchmark and the zero-shot accuracy on the GSM8K benchmark in Table 2. On the IFEval benchmark, PMRC achieves the best performance among all projection-based methods across the three models, outperforming GUM by more than 2 percentage points. Notably, on Qwen3-8B, PMRC also surpasses the full-space optimizers. On the GSM8K benchmark, PMRC achieves the best performance among all memory-efficient methods while remaining competitive with the full-space optimizers.

7 CONCLUSION

In this paper, we investigate memory-efficient subspace optimization and adopt a residual correction framework. For momentum SGD, our proposed SMRC combines projected momentum with a gradient correction in the orthogonal complement and achieves an $O(T^{-1/4})$ convergence rate, which matches that of full-space momentum SGD. We generalize SMRC to a multi-stage variant with arbitrary basis refreshes and introduce a boundary correction that aligns consecutive virtual iterates, while retaining a convergence rate of the same order as the single-stage method. Furthermore, we extend this framework to Muon and propose PMRC, which achieves an $O(T^{-1/4})$ rate under a mixed stationarity measure, while the multi-stage variant retains a convergence rate of the same order. The convergence rates of both PMRC variants are of the same order as that of full-space Muon. Both SMRC and PMRC reduce the optimizer-state memory from $O(mn)$ to $O((m+n)r)$. Extensive experimental results on LLM pretraining and supervised finetuning tasks demonstrate the practical effectiveness and efficiency of our proposed methods.

AI USE STATEMENT

In this work, we used generative AI tools for polishing, assistance with checking mathematical proofs, creating or editing software code, suggesting experimental parameters, proof organization, and scripts for plotting results. We reviewed all AI-assisted content. LLM-generated code was verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All theoretical claims are stated with their required assumptions in the main text, and complete proofs are provided in the appendix. The experimental setup, hyperparameters, and training details are fully described in the appendix. The code for reproducing experiments will be made publicly available upon acceptance.

ETHICS STATEMENT

This work is methodological and does not involve human subjects, personal data, or applications that raise specific ethical concerns. We are not aware of any ethical issues beyond standard considerations of research integrity and responsible reporting.

REFERENCES

Kang An, Yuxing Liu, Rui Pan, Yi Ren, Shiqian Ma, Donald Goldfarb, and Tong Zhang. ASGO: Adaptive structured gradient optimization. In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, pp. 126775–126814, 2025.

Ben Anson, Conor Houghton, and Edward Milsom. Approximate Muon with low-rank adapters. *arXiv preprint arXiv:2608.14492*, 2026.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2022.

Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. *arXiv preprint arXiv:2409.20325*, 2024.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 7432–7439, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 1877–1901, 2020.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

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 *Advances in Neural Information Processing Systems 38 (NeurIPS)*, 2025a.

Yiming Chen, Yuan Zhang, Yin Liu, Kun Yuan, and Zaiwen Wen. A memory efficient randomized subspace optimization method for training large language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025b.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240): 1–113, 2023.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*, 2021.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, pp. 10088–10115, 2023.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023.

Takeshi Fukaya, Ramaseshan Kannan, Yuji Nakatsukasa, Yusaku Yamamoto, and Yuka Yanagisawa. Shifted cholesky qr for computing the qr factorization of ill-conditioned matrices. *SIAM Journal on Scientific Computing*, 42(1), 2020.

Athanasios Glentis, Jiaxiang Li, Andi Han, and Mingyi Hong. Memory-efficient llm pretraining via minimalist optimizer design. *arXiv preprint arXiv:2506.16659*, 2025.

Wenbo Gong, Javier Zazo, Qijun Luo, Puqian Wang, James Hensman, and Chao Ma. ARO: A new lens on matrix optimization for large models. *arXiv preprint arXiv:2602.09006*, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, et al. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Aman Gupta, Rafael Celente, Abhishek Shivanna, D. T. Braithwaite, Gregory Dexter, Shao Tang, Hiroto Udagawa, Daniel Silva, Rohan Ramanath, and S. Sathiya Keerthi. Effective quantization of Muon optimizer states. *arXiv preprint arXiv:2509.23106*, 2025.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 1842–1850, 2018.

Andi Han, Jiaxiang Li, Wei Huang, Mingyi Hong, Akiko Takeda, Pratik 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 (NeurIPS)*, pp. 118267–118295, 2024.

Yongchang Hao, Yanshuai Cao, and Lili Mou. Flora: Low-rank adapters are secretly gradient compressors. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 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 *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pp. 22468–22522, 2025.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pp. 30016–30030, 2022.

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 (ICML)*, 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 (ICLR)*, 2022.

Feihu Huang, Yuning Luo, and Songcan Chen. LiMuon: Light and fast Muon optimizer for large models. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Wei Jiang, Jiayu Qin, Lingyu Wu, Changyou Chen, Tianbao Yang, and Lijun Zhang. Learning unnormalized statistical models via compositional optimization. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 15105–15124, 2023.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL `https://kellerjordan.github.io/posts/muon/`.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *The Third International Conference on Learning Representations (ICLR)*, 2015.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pp. 3843–3857, 2022.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions, 2024.

Jiaxiang Li and Mingyi Hong. A note on the convergence of Muon. *arXiv preprint arXiv:2502.02900*, 2025.

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 (ICLR)*, 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 37 (NeurIPS)*, pp. 64412–64432, 2024.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for LLM training. *arXiv preprint arXiv:2502.16982*, 2025.

Yanli Liu, Yuan Gao, and Wotao Yin. An improved analysis of stochastic gradient descent with momentum. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 18261–18271, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *The Seventh International Conference on Learning Representations (ICLR)*, 2019.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing (EMNLP)*, pp. 2381–2391, 2018.

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

Thien Hang Nguyen and Huy Le Nguyen. Lean and mean adaptive optimization via subset-norm and subspace-momentum with convergence guarantees. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pp. 46116–46161, 2025.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 OLMo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Rui Pan, Xiang Liu, Shizhe Diao, Renjie Pi, Jipeng Zhang, Chi Han, and Tong Zhang. LISA: Layerwise importance sampling for memory-efficient large language model fine-tuning. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, pp. 57018–57049, 2024.

Rui Pan, Yang Luo, Yuxing Liu, Yang You, and Tong Zhang. Unbiased gradient low-rank projection. *arXiv preprint arXiv:2510.17802*, 2025a.

Rui Pan, Dylan Zhang, Hanning Zhang, Xingyuan Pan, Minrui Xu, Jipeng Zhang, Renjie Pi, Xiaoyu Wang, and Tong Zhang. ScaleBiO: Scalable bilevel optimization for LLM data reweighting. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31959–31982, 2025b.

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.

Yehonathan Refael, Guy Smorodinsky, Tom Tirer, and Ofir Lindenbaum. Sumo: Subspace-aware moment-orthogonalization for accelerating memory-efficient llm training. In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, pp. 147250–147281, 2025.

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 (ICLR)*, 2025.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 4463–4473, 2019.

Yupeng Su, Ruijie Zhang, Ziyue Liu, Yequan Zhao, and Zheng Zhang. Muonq: Enhancing low-bit muon quantization via directional fidelity optimization. *arXiv preprint arXiv:2605.11396*, 2026.

Kimi Team, Yifan Bai, Yiping Bao, Y Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. LaMDA: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, 2017.

Shuo Xie, Tianhao Wang, Sashank J. Reddi, Sanjiv Kumar, and Zhiyuan Li. Structured preconditioners in adaptive optimization: A unified analysis. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pp. 68746–68776, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025.

Dingzhi Yu, Rui Pan, Yuxing Liu, Difan Zou, and Tong Zhang. StoSignSGD: Unbiased structural stochasticity fixes SignSGD for training large language models. *arXiv preprint arXiv:2604.15416*, 2026a.

Dingzhi Yu, Hongyi Tao, Yuanyu Wan, Luo Luo, and Lijun Zhang. Sign-based optimizers are effective under heavy-tailed noise. *arXiv preprint arXiv:2602.07425*, 2026b.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

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 (NeurIPS)*, pp. 10122–10174, 2025a.

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 (ICLR)*, 2023.

Yushun Zhang, Congliang Chen, Ziniu Li, Tian Ding, Chenwei Wu, Diederik P Kingma, Yinyu Ye, Zhi-Quan Luo, and Ruoyu Sun. Adam-mini: Use fewer learning rates to gain more. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025b.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. GaLore: Memory-efficient LLM training by gradient low-rank projection. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pp. 61121–61143, 2024.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-Following Evaluation for Large Language Models. *arXiv preprint arXiv:2311.07911*, 2023.

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.

Philip Zmushko, Aleksandr Beznosikov, Martin Takáč, and Samuel Horváth. FRUGAL: Memory-efficient optimization by reducing state overhead for scalable training. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pp. 80708–80739, 2025.

A Additional Discussion on Related Work

A.1 Additional Memory-Efficient Optimization Methods

Several methods reduce memory cost through structured restrictions on the trainable parameters. Han et al. (2024) combine sparse and low-rank trainable components to increase the expressiveness of memory-efficient updates. LISA (Pan et al., 2024) periodically resamples a subset of intermediate layers and updates parameters within the selected layers.

Another line of work improves the construction and evolution of low-dimensional optimization subspaces. Flora (Hao et al., 2024) repeatedly resamples random low-rank projections, allowing updates from different subspaces to accumulate into higher-rank parameter changes. Muhamed et al. (2024) construct sparse projection matrices from gradient row norms, which reduces the computational cost relative to GaLore. Liang et al. (2024) instead maintain the projection basis through online subspace estimation. SARA (Zhang et al., 2025a) samples singular directions according to their importance rather than deterministically retaining only the leading directions. Chen et al. (2025b) formulate training over sampled low-dimensional parameter subspaces, reducing optimizer-state and activation memory.

Comparison with Nguyen & Nguyen (2025). Nguyen & Nguyen (2025) combine projected momentum with an SGD update in the orthogonal complement. Their convergence guarantee is established for a fixed projection under stronger assumptions, with the residual update restricted to unit weight. It does not provide guarantees for subspace switching or the matrix-sign updates used by Muon. We address these limitations by analyzing weighted residual correction, establishing convergence for multi-stage SMRC with boundary correction, and developing a mixed-geometry analysis for PMRC. Moreover, the proof of Lemma D.1 in Nguyen & Nguyen (2025) contains an error. In our notation with $\gamma = 1$ and $\mathbf{m}_0 = \mathbf{0}$, the argument relies on the following inequalities

$$
\begin{aligned}
&-\eta \sum_{t=1}^{T} \left\langle P^\top \nabla F(W_t), \mathbf{m}_t \right\rangle \\
\leq & L\eta^2 \sum_{t=1}^{T} \sum_{i=1}^{t-1} (1-\beta)^{t-i} \left\| P\mathbf{m}_i + QG_i \right\|_F^2 - \beta\eta \sum_{t=1}^{T} \sum_{i=1}^{t} (1-\beta)^{t-i} \left\langle P^\top \nabla F(W_i), P^\top G_i \right\rangle \\
= & L\eta^2 \sum_{i=1}^{T-1} \left\| P\mathbf{m}_i + QG_i \right\|_F^2 \sum_{t=i+1}^{T} (1-\beta)^{t-i} - \beta\eta \sum_{i=1}^{T} \left\langle P^\top \nabla F(W_i), P^\top G_i \right\rangle \sum_{t=i}^{T} (1-\beta)^{t-i} \\
\leq & \frac{L\eta^2}{\beta} \sum_{t=1}^{T} \left\| P\mathbf{m}_t + QG_t \right\|_F^2 - \eta \sum_{t=1}^{T} \left\langle P^\top \nabla F(W_t), P^\top G_t \right\rangle .
\end{aligned}
$$

It omits the term $\eta \sum_{t=1}^{T} (1-\beta)^{T-t+1} \left\langle P^\top \nabla F(W_t), P^\top G_t \right\rangle$. When $G_t = \nabla F(W_t)$, this quantity reduces to $\eta \sum_{t=1}^{T} (1-\beta)^{T-t+1} \left\| P^\top \nabla F(W_t) \right\|_F^2 \geq 0$. Its removal can make the right-hand side smaller and is not justified by the bound in the original paper.

Other methods reduce memory by directly compressing the optimizer states. Adam-mini (Zhang et al., 2025b) replaces elementwise second-moment estimates with a scalar estimate shared within each Hessian-informed parameter block, while retaining the full first-moment estimate. APOLLO (Zhu et al., 2025) estimates channel-wise gradient scaling from low-rank first- and second-moment states constructed using random projections.

A.2 Low-rank Adaptation (LoRA)

Low-Rank Adaptation (LoRA) (Hu et al., 2022) is an efficient finetuning method for LLMs. LoRA reduces the memory cost of finetuning by freezing the pretrained weights and training low-rank adapter matrices. Subsequent variants extend this approach through adaptive rank allocation across weight matrices (Zhang et al., 2023), periodic merging of low-rank updates into the base weights (Lialin et al., 2024), and quantization of the frozen weights (Dettmers et al., 2023).

A.3 Matrix-Aware and Muon-Based Optimizers

Matrix-aware optimization exploits the structure of model parameters. Shampoo (Gupta et al., 2018) maintains factored preconditioners along individual dimensions. Xie et al. (2025) and An et al. (2025) further construct adaptive preconditioners from structured matrix gradients.

A separate line of work focuses on reducing the memory cost of Muon. SUMO (Refael et al., 2025) orthogonalizes the momentum within a dynamically adapted low-dimensional subspace, but its convergence guarantee is established only for reversible layers. LiMuon (Huang et al., 2026) combines momentum-based variance reduction with randomized SVD to compress the stored gradient estimate, reducing the memory required for optimizer states. Gupta et al. (2025) reduce the memory cost of Muon through blockwise 8-bit quantization, while Su et al. (2026) further compress the optimizer states to 4-bit precision.

B Proofs of the Theorems

B.1 Proof of Theorem 1

Throughout the proof, let $\{\mathcal{F}_t\}_{t=0}^T$ denote the natural filtration, where $\mathcal{F}_{t-1}$ contains all randomness revealed before the stochastic gradient G_t is queried. We define $Q = I - PP^\top$ and $\mathbf{v}_t = P\mathbf{m}_t$. According to the momentum update of Algorithm 1, we have

$$\mathbf{v}_t = (1-\beta)\mathbf{v}_{t-1} + \beta\, PP^\top G_t, \quad \mathbf{v}_0 = \mathbf{0}. \tag{1}$$

We denote $c_\beta = \frac{1-\beta}{\beta}$ and define the virtual iterate

$$U_t = W_t - c_\beta \eta\, \mathbf{v}_{t-1}, \tag{2}$$

so that $U_1 = W_1$. Then we define

$$C = PP^\top + \gamma Q = \gamma I + (1-\gamma)PP^\top. \tag{3}$$

Since $PP^\top$ and Q are complementary orthogonal projectors, C is symmetric with eigenvalue 1 of multiplicity r and eigenvalue γ of multiplicity $m-r$. In particular, $0 \prec \gamma I \preceq C \preceq I$ and $C^2 \preceq C$ for $\gamma \in (0,1]$.

By substituting the update $W_{t+1} = W_t - \eta(\mathbf{v}_t + \gamma Q G_t)$ into Eq. (2), we can derive

$$U_{t+1} = W_{t+1} - c_\beta \eta\, \mathbf{v}_t = W_t - \eta(1+c_\beta)\mathbf{v}_t - \eta\gamma Q G_t. \tag{4}$$

Since $1 + c_\beta = 1/\beta$, the recursion Eq. (1) implies

$$(1+c_\beta)\mathbf{v}_t = \beta^{-1}\mathbf{v}_t = c_\beta\, \mathbf{v}_{t-1} + PP^\top G_t. \tag{5}$$

By substituting Eq. (5) into Eq. (4) and using Eq. (3), we obtain

$$U_{t+1} = (W_t - c_\beta \eta\, \mathbf{v}_{t-1}) - \eta\big(PP^\top + \gamma Q\big)G_t = U_t - \eta\, C G_t. \tag{6}$$

By using Assumption 1 and Eq. (6), we have

$$F(U_{t+1}) \le F(U_t) - \eta\, \langle \nabla F(U_t), C G_t \rangle + \frac{L\eta^2}{2} \|C G_t\|_F^2 .$$

Furthermore, we have

$$\mathbb{E}\left[F(U_{t+1}) \mid \mathcal{F}_{t-1}\right] \le F(U_t) - \eta\, \langle \nabla F(U_t), C\, \nabla F(W_t) \rangle + \frac{L\eta^2}{2}\, \mathbb{E}\left[\|C\, G_t\|_F^2 \mid \mathcal{F}_{t-1}\right]. \tag{7}$$

We denote $\nabla F(U_t) = \nabla F(W_t) + \Delta_t$, where $\Delta_t = \nabla F(U_t) - \nabla F(W_t)$, and define

$$S_t = \langle \nabla F(W_t), C \nabla F(W_t) \rangle = \|PP^\top \nabla F(W_t)\|_F^2 + \gamma \|Q \nabla F(W_t)\|_F^2.$$

Then we can derive

$$\langle \nabla F(U_t), C\, \nabla F(W_t) \rangle = S_t + \langle \Delta_t, C\, \nabla F(W_t) \rangle . \tag{8}$$

By using Assumption 1, we have

$$\|\Delta_t\|_F \leq L\|U_t - W_t\|_F = Lc_\beta\eta\,\|\mathbf{v}_{t-1}\|_F.$$

Moreover, since $\gamma \in (0, 1]$, we have $\|C\|_2 = 1$. Therefore, we can derive

$$\big|\langle \Delta_t, C\nabla F(W_t)\rangle\big| \leq \|\Delta_t\|_F\,\|C\nabla F(W_t)\|_F \leq Lc_\beta\eta\,\|\mathbf{v}_{t-1}\|_F\,\|C\nabla F(W_t)\|_F. \tag{9}$$

By using $C^2 \preceq C$, we have $\|C\nabla F(W_t)\|_F^2 \leq S_t$. Applying Young's inequality, we can further derive

$$\big|\langle \Delta_t, C\nabla F(W_t)\rangle\big| \leq \frac{1}{4}S_t + L^2{c_\beta}^2\eta^2\|\mathbf{v}_{t-1}\|_F^2. \tag{10}$$

Combining Eq. (8) and Eq. (10), we can obtain

$$\langle \nabla F(U_t), C\,\nabla F(W_t)\rangle \geq \frac{3}{4}S_t - L^2{c_\beta}^2\eta^2\|\mathbf{v}_{t-1}\|_F^2. \tag{11}$$

Moreover, since $C^2 \preceq C \preceq I$, we have $\|C\nabla F(W_t)\|_F^2 \leq S_t$ and $\mathbb{E}[\|C(G_t - \nabla F(W_t))\|_F^2] \leq \sigma^2$. Therefore, we obtain

$$\begin{aligned}\mathbb{E}\left[\|C\,G_t\|_F^2 \mid \mathcal{F}_{t-1}\right] &\leq \|C\nabla F(W_t)\|_F^2 + \mathbb{E}\left[\|C(G_t - \nabla F(W_t))\|_F^2 \mid \mathcal{F}_{t-1}\right] \\ &\leq S_t + \sigma^2.\end{aligned} \tag{12}$$

Plugging Eq. (11) and Eq. (12) into Eq. (7), we can derive

$$\mathbb{E}[F(U_{t+1})] \leq \mathbb{E}[F(U_t)] - \eta\Big(\frac{3}{4} - \frac{L\eta}{2}\Big)\mathbb{E}[S_t] + L^2{c_\beta}^2\eta^3\mathbb{E}\left[\|\mathbf{v}_{t-1}\|_F^2\right] + \frac{L\eta^2}{2}\sigma^2.$$

By setting $\eta \leq \gamma/(2L) \leq 1/(2L)$, we have $\frac{3}{4} - \frac{L\eta}{2} \geq \frac{1}{2}$ and can ensure

$$\mathbb{E}[F(U_{t+1})] \leq \mathbb{E}[F(U_t)] - \frac{\eta}{2}\mathbb{E}[S_t] + L^2{c_\beta}^2\eta^3\mathbb{E}\left[\|\mathbf{v}_{t-1}\|_F^2\right] + \frac{L\eta^2}{2}\sigma^2.$$

By summing up over $t = 1, \ldots, T$, we have

$$\frac{\eta}{2}\sum_{t=1}^{T}\mathbb{E}\left[S_t\right] \leq \mathbb{E}[F(W_1)] - \mathbb{E}[F(U_{T+1})] + L^2{c_\beta}^2\eta^3\sum_{t=1}^{T}\mathbb{E}\left[\|\mathbf{v}_{t-1}\|_F^2\right] + \frac{L\eta^2}{2}T\sigma^2. \tag{13}$$

It remains to bound the term $\sum_{t=1}^{T}\mathbb{E}\left[\|\mathbf{v}_{t-1}\|_F^2\right]$. By using Eq. (1), we have

$$\|\mathbf{v}_t\|_F^2 = \left\|(1-\beta)\mathbf{v}_{t-1} + \beta\,PP^\top G_t\right\|_F^2 \leq (1-\beta)\|\mathbf{v}_{t-1}\|_F^2 + \beta\|PP^\top G_t\|_F^2,$$

which further implies $\beta\|\mathbf{v}_{t-1}\|_F^2 \leq \|\mathbf{v}_{t-1}\|_F^2 - \|\mathbf{v}_t\|_F^2 + \beta\|PP^\top G_t\|_F^2$. Summing over $t = 1, \ldots, T$ and using $\mathbf{v}_0 = \mathbf{0}$, we can derive $\sum_{t=1}^{T}\mathbb{E}\left[\|\mathbf{v}_{t-1}\|_F^2\right] \leq \sum_{t=1}^{T}\mathbb{E}\left[\|PP^\top G_t\|_F^2\right]$.

According to Assumption 3 and $\|PP^\top\|_2 \leq 1$, we have

$$\sum_{t=1}^{T}\mathbb{E}\left[\|PP^\top G_t\|_F^2\right] \leq \sum_{t=1}^{T}\mathbb{E}\left[\|PP^\top\nabla F(W_t)\|_F^2\right] + T\sigma^2 \leq \sum_{t=1}^{T}\mathbb{E}\left[S_t\right] + T\sigma^2.$$

Finally, substituting these two inequalities into Eq. (13), we obtain

$$\Big(\frac{\eta}{2} - L^2{c_\beta}^2\eta^3\Big)\sum_{t=1}^{T}\mathbb{E}[S_t] \leq \mathbb{E}[F(W_1)] - \mathbb{E}[F(U_{T+1})] + \Big(L^2{c_\beta}^2\eta^3 + \frac{L\eta^2}{2}\Big)T\sigma^2. \tag{14}$$

By setting $\eta \leq \gamma/(2Lc_\beta) \leq 1/(2Lc_\beta)$ when $c_\beta > 0$, we have $L^2{c_\beta}^2\eta^2 \leq \frac{1}{4}$ and $\frac{\eta}{2} - L^2{c_\beta}^2\eta^3 \geq \frac{\eta}{4}$. Therefore, Eq. (14) implies

$$\begin{aligned}\frac{\eta}{4}\sum_{t=1}^{T}\mathbb{E}\left[S_t\right] &\leq \mathbb{E}[F(W_1)] - \mathbb{E}[F(U_{T+1})] + \Big(L^2{c_\beta}^2\eta^3 + \frac{L\eta^2}{2}\Big)T\sigma^2 \\ &\leq \mathbb{E}[F(W_1)] - F_* + \Big(L^2{c_\beta}^2\eta^3 + \frac{L\eta^2}{2}\Big)T\sigma^2,\end{aligned} \tag{15}$$

which further implies

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\Big[\big\|PP^\top\nabla F(W_t)\big\|_F^2+\gamma\big\|Q\nabla F(W_t)\big\|_F^2\Big]\le\frac{4\big(\mathbb{E}[F(W_1)]-F_*\big)}{\eta\,T}+\big(2L\eta+4L^2{c_\beta}^2\eta^2\big)\sigma^2.$$

Since $S_t\ge\gamma\|\nabla F(W_t)\|_F^2$ and $\eta=\Theta(T^{-1/2})$ for fixed β and γ, we obtain

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\Big[\big\|\nabla F(W_t)\big\|_F^2\Big]\le O(T^{-1/2}).$$

Finally, applying Jensen's inequality yields

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla F(W_t)\|_F\right]\le\left(\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla F(W_t)\|_F^2\right]\right)^{1/2}=O(T^{-1/4}).$$

B.2 Proof of Theorem 2

For each stage $k\in\{1,\dots,K\}$, let $\{\mathcal{F}_{k,t}\}_{t=0}^{T_k}$ denote the filtration, with $\mathcal{F}_{k,0}$ corresponding to the optimization history prior to stage k. By Assumption 4, P_k is $\mathcal{F}_{k,0}$-measurable and hence $\mathcal{F}_{k,t-1}$-measurable for every $t\in\{1,\dots,T_k\}$. Define $Q_k=I-P_kP_k^\top$, $\mathbf{v}_{k,t}=P_k\mathbf{m}_{k,t}$ and the virtual iterate $U_{k,t}=W_{k,t}-c_\beta\eta\,\mathbf{v}_{k,t-1}$.

Since the momentum is reset at the start of each stage, i.e., $\mathbf{m}_{k,0}=\mathbf{0}$, we have $\mathbf{v}_{k,0}=\mathbf{0}$ and $U_{k,1}=W_{k,1}$. We define $S_{k,t}=\big\|P_kP_k^\top\nabla F(W_{k,t})\big\|_F^2+\gamma\big\|Q_k\nabla F(W_{k,t})\big\|_F^2$.

Stage k performs T_k iterations of Algorithm 1, initialized at $W_{k,1}$ with $\mathbf{m}_{k,0}=\mathbf{0}$. By setting $\eta\le\min\left\{\frac{\gamma}{2L},\frac{\gamma}{2Lc_\beta},\frac{1}{L\sqrt{N}}\right\}$, the single-stage inequality Eq. (15) applies with (W_t,G_t,U_t,T) replaced by $(W_{k,t},G_{k,t},U_{k,t},T_k)$. Taking expectations, we have

$$\frac{\eta}{4}\sum_{t=1}^{T_k}\mathbb{E}\left[S_{k,t}\right]\le\mathbb{E}\big[F(W_{k,1})\big]-\mathbb{E}\big[F(U_{k,T_k+1})\big]+\Big(L^2{c_\beta}^2\eta^3+\frac{L\eta^2}{2}\Big)T_k\sigma^2.\tag{16}$$

By summing Eq. (16) over $k=1,\dots,K$ and defining $N=\sum_{k=1}^{K}T_k$, we have

$$\frac{\eta}{4}\sum_{k=1}^{K}\sum_{t=1}^{T_k}\mathbb{E}[S_{k,t}]\le\sum_{k=1}^{K}\Big(\mathbb{E}[F(W_{k,1})]-\mathbb{E}[F(U_{k,T_k+1})]\Big)+\Big(L^2{c_\beta}^2\eta^3+\frac{L\eta^2}{2}\Big)N\sigma^2.\tag{17}$$

Since we set $W_{k+1,1}=W_{k,T_k+1}-c_\beta\eta\,P_k\mathbf{m}_{k,T_k}$, we can ensure $W_{k+1,1}=U_{k,T_k+1}$.

Consequently, we further derive

$$\sum_{k=1}^{K}\Big(\mathbb{E}[F(W_{k,1})]-\mathbb{E}[F(U_{k,T_k+1})]\Big)=\mathbb{E}[F(W_{1,1})]-\mathbb{E}[F(U_{K,T_K+1})]\le F(W_{1,1})-F_*,\tag{18}$$

where the inequality uses Assumption 2.

By combining Eq. (17) and Eq. (18), we have

$$\frac{1}{N}\sum_{k=1}^{K}\sum_{t=1}^{T_k}\mathbb{E}\left[S_{k,t}\right]\le\frac{4\left(\mathbb{E}\left[F(W_{1,1})\right]-F_*\right)}{\eta N}+\left(2L\eta+4L^2{c_\beta}^2\eta^2\right)\sigma^2.$$

By further using $S_{k,t}\ge\gamma\|\nabla F(W_{k,t})\|_F^2$, we have

$$\frac{1}{N}\sum_{k=1}^{K}\sum_{t=1}^{T_k}\mathbb{E}\left[\big\|\nabla F(W_{k,t})\big\|_F^2\right]\le\frac{4\big(\mathbb{E}[F(W_{1,1})]-F_*\big)}{\gamma\,\eta\,N}+\Big(\frac{2L\eta}{\gamma}+\frac{4L^2{c_\beta}^2\eta^2}{\gamma}\Big)\sigma^2.$$

With fixed β and γ, the choice $\eta = \min\{1/(L\sqrt{N}), \gamma/(2L), \gamma/(2Lc_\beta)\}$ satisfies $\eta = \Theta(N^{-1/2})$. Finally, applying Jensen's inequality yields

$$\frac{1}{N}\sum_{k=1}^{K}\sum_{t=1}^{T_k}\mathbb{E}\left[\|\nabla F(W_{k,t})\|_F\right] \le \left(\frac{1}{N}\sum_{k=1}^{K}\sum_{t=1}^{T_k}\mathbb{E}\left[\|\nabla F(W_{k,t})\|_F^2\right]\right)^{1/2} = O(N^{-1/4}),$$

which completes the proof.

B.3 Proof of Theorem 3

We first present several definitions. Let $\Delta = F(W_1) - F_*$ denote the initial optimality gap. For $X \in \mathbb{R}^{r\times n}$, $\mathrm{msign}(X)$ is equal to the solution of the problem $\arg\max_{\|Z\|_2\le 1}\langle X, Z\rangle$. Hence we have

$$\|\mathrm{msign}(X)\|_2 \le 1, \quad \|\mathrm{msign}(X)\|_F^2 \le r, \quad \langle X, \mathrm{msign}(X)\rangle = \|X\|_*.$$

We define $A_t = P^\top \nabla F(W_t)$, $R_t = Q\nabla F(W_t)$ and $\xi_t = G_t - \nabla F(W_t)$. The corrected momentum and the corresponding tracking error are

$$\tilde{M}_t = \frac{M_t}{1-(1-\beta)^t}, \quad \tilde{E}_t = \tilde{M}_t - A_t, \quad t \ge 1.$$

We further define

$$\bar{A} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|P^\top\nabla F(W_t)\right\|_*\right], \qquad \bar{R} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|Q\nabla F(W_t)\|_F\right],$$
$$\bar{S} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|Q\nabla F(W_t)\|_F^2\right], \qquad \overline{\mathcal{E}} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\tilde{E}_t\right\|_*\right].$$

Then we introduce several useful lemmas.

Lemma 1 (Lemma D.2 in Yu et al. (2026b)) *For any $X \in \mathbb{R}^{m\times n}$ and $a > 0$, the following holds*

1. $\langle X, \mathrm{msign}(X)\rangle = \|X\|_*$,
2. $\mathrm{msign}(aX) = \mathrm{msign}(X)$.

Lemma 2 (Lemma D.3 in Yu et al. (2026b)) *For any $A, B \in \mathbb{R}^{r\times n}$ and any $Z_B = \mathrm{msign}(B)$,*

$$\langle A, Z_B\rangle \ge \|A\|_* - 2\|A - B\|_*.$$

Lemma 3 *Let $a = 1-\beta$ and $\omega_{i,t} = \frac{\beta a^{t-i}}{1-a^t}, 1 \le i \le t$. Then we have*

$$\tilde{M}_t = \sum_{i=1}^{t}\omega_{i,t}P^\top G_i, \quad \sum_{i=1}^{t}\omega_{i,t} = 1, \quad v_t = \sum_{i=1}^{t}\omega_{i,t}^2 = \frac{\beta}{2-\beta}\frac{1+a^t}{1-a^t}.$$

For $H \ge 1$, we have $\frac{1}{H}\sum_{t=1}^{H}\sqrt{v_t} \le 2\sqrt{2}\left(\sqrt{\beta} + H^{-1/2}\right)$. Moreover, for every $t \ge 1$, we have

$$\sum_{i=1}^{t}\omega_{i,t}(t-i) \le \frac{1}{\beta}, \qquad \sum_{i=1}^{t}\omega_{i,t}\sqrt{t-i} \le \frac{1}{\sqrt{\beta}}.$$

Lemma 4 *For each $Z \in \mathbb{R}^{r\times n}$ and $G \in \mathbb{R}^{m\times n}$, we have*

$$\langle PZ, QG\rangle = 0, \qquad \|PZ + \gamma QG\|_F^2 = \|Z\|_F^2 + \gamma^2\|QG\|_F^2.$$

In particular, if $Z = \mathrm{msign}(X)$ for some $X \in \mathbb{R}^{r\times n}$, then we have

$$\|PZ + \gamma QG\|_F^2 \le r + \gamma^2\|QG\|_F^2.$$

We start the proof by establishing a bound on $\bar{A} + \frac{\gamma}{2}\bar{S}$.

Lemma 5 *Under Assumptions 1, 2 and 3, if $L\eta\gamma \leq 1$, we have*

$$\bar{A} + \frac{\gamma}{2}\bar{S} \leq \frac{F(W_1) - F_*}{\eta T} + 2\overline{\mathcal{E}} + \frac{L\eta r}{2} + \frac{L\eta\gamma^2\sigma^2}{2}.$$

Moreover, for each $t = 1, \ldots, T$, we have

$$\eta\mathbb{E}\left[\|A_t\|_*\right] + \frac{\eta\gamma}{2}\mathbb{E}\left[\|R_t\|_F^2\right] \leq \mathbb{E}\big[F(W_t) - F(W_{t+1})\big] + 2\eta\mathbb{E}\left[\left\|\tilde{E}_t\right\|_*\right] + \frac{L\eta^2 r}{2} + \frac{L\eta^2\gamma^2\sigma^2}{2}.$$

It remains to derive a bound on $\overline{\mathcal{E}}$ and we present the following lemma.

Lemma 6 *Under Assumptions 1, 2 and 3, we have*

$$\overline{\mathcal{E}} \leq 2\sqrt{2r}\,\sigma\left(\sqrt{\beta} + T^{-1/2}\right) + \frac{Lr\eta}{\beta} + \frac{L\sqrt{r}\,\eta\gamma\sigma}{\sqrt{\beta}} + \frac{L\sqrt{r}\,\eta\gamma}{\beta}\bar{R}.$$

By combining Lemma 5 with Lemma 6, we have

$$\begin{aligned}\bar{A} + \frac{\gamma}{2}\bar{S} \leq \frac{\Delta}{\eta T} + 4\sqrt{2r}\,\sigma\left(\sqrt{\beta} + T^{-1/2}\right) + \frac{2Lr\eta}{\beta} + \frac{2L\sqrt{r}\,\eta\gamma\sigma}{\sqrt{\beta}} \\ + \frac{2L\sqrt{r}\,\eta\gamma}{\beta}\bar{R} + \frac{L\eta r}{2} + \frac{L\eta\gamma^2\sigma^2}{2}.\end{aligned}$$

Then we derive a bound on $\bar{R}$. By using Jensen's inequality and Young's inequality, we have

$$\frac{2L\sqrt{r}\,\eta\gamma}{\beta}\bar{R} \leq \frac{\gamma}{4}\bar{R}^2 + \frac{4L^2 r\eta^2\gamma}{\beta^2} \leq \frac{\gamma}{4}\bar{S} + \frac{4L^2 r\eta^2\gamma}{\beta^2},$$

which further implies

$$\begin{aligned}\bar{A} + \frac{\gamma}{4}\bar{S} \leq \frac{\Delta}{\eta T} + 4\sqrt{2r}\,\sigma\left(\sqrt{\beta} + T^{-1/2}\right) + \frac{2Lr\eta}{\beta} + \frac{2L\sqrt{r}\,\eta\gamma\sigma}{\sqrt{\beta}} \\ + \frac{L\eta r}{2} + \frac{L\eta\gamma^2\sigma^2}{2} + \frac{4L^2 r\eta^2\gamma}{\beta^2}.\end{aligned} \tag{19}$$

We set $\beta = T^{-1/2}, \eta = \frac{1}{2L}T^{-3/4}, \gamma = T^{1/4}$, which satisfy $\gamma\eta L \leq 1$. According to Eq. (19), we have

$$\bar{A} + \frac{\gamma}{4}\bar{S} \leq \left[2L\Delta + \frac{9r}{4} + \frac{\sigma^2}{4} + (8\sqrt{2}+1)\sqrt{r}\,\sigma\right]T^{-1/4}.$$

To derive the guarantee for unsquared stationarity, we introduce a useful inequality. For each $x \geq 0$ and $\gamma > 0$, we have

$$x \leq \frac{\gamma}{4}x^2 + \frac{1}{\gamma}. \tag{20}$$

By using Eq. (20), we can derive $\bar{R} \leq \frac{\gamma}{4}\bar{S} + \frac{1}{\gamma} = \frac{\gamma}{4}\bar{S} + T^{-1/4}$.

Therefore, we can obtain

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|P^\top\nabla F(W_t)\right\|_* + \left\|Q\nabla F(W_t)\right\|_F\right] \leq \left[2L\Delta + \frac{9r}{4} + \frac{\sigma^2}{4} + (8\sqrt{2}+1)\sqrt{r}\,\sigma + 1\right]T^{-1/4}.$$

Moreover, by using $\|X\|_F \leq \left\|P^\top X\right\|_* + \|QX\|_F$, we can derive

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla F(W_t)\|_F\right] \leq \left[2L\Delta + \frac{9r}{4} + \frac{\sigma^2}{4} + (8\sqrt{2}+1)\sqrt{r}\,\sigma + 1\right]T^{-1/4}.$$

B.4 Proof of Theorem 4

First, we present several definitions. Let $\Delta = F(W_{1,1}) - F_*$ denote the initial optimality gap. Consider K stages of length T. The total number of gradient evaluations is $N = KT$. At the start of stage k, the algorithm selects a predictable orthonormal basis $P_k \in \mathbb{R}^{m\times r}$ and sets $Q_k = I - P_k P_k^\top$, $M_{k,0} = \mathbf{0}$. We define

$$A_{k,t} = P_k^\top \nabla F(W_{k,t}), \quad R_{k,t} = Q_k \nabla F(W_{k,t}), \quad \tilde{M}_{k,t} = \frac{M_{k,t}}{1-(1-\beta)^t}, \quad \tilde{E}_{k,t} = \tilde{M}_{k,t} - A_{k,t}.$$

$$\bar{A} = \frac{1}{N}\sum_{k=1}^{K}\sum_{t=1}^{T}\mathbb{E}\left[\|A_{k,t}\|_*\right], \qquad \bar{R} = \frac{1}{N}\sum_{k=1}^{K}\sum_{t=1}^{T}\mathbb{E}\left[\|R_{k,t}\|_F\right],$$
$$\bar{S} = \frac{1}{N}\sum_{k=1}^{K}\sum_{t=1}^{T}\mathbb{E}\left[\|R_{k,t}\|_F^2\right], \qquad \overline{\mathcal{E}} = \frac{1}{N}\sum_{k=1}^{K}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\tilde{E}_{k,t}\right\|_*\right].$$

As in the previous proof, we first establish the guarantee for multi-stage descent. We can apply Lemma 5 and replace $(P, Q, W_t, G_t, A_t, R_t, \tilde{E}_t)$ by $(P_k, Q_k, W_{k,t}, G_{k,t}, A_{k,t}, R_{k,t}, \tilde{E}_{k,t})$. Thus, we can derive

$$\eta\mathbb{E}\left[\|A_{k,t}\|_*\right] + \frac{\eta\gamma}{2}\mathbb{E}\left[\|R_{k,t}\|_F^2\right] \le \mathbb{E}\big[F(W_{k,t}) - F(W_{k,t+1})\big] + 2\eta\mathbb{E}\left[\left\|\tilde{E}_{k,t}\right\|_*\right] + \frac{L\eta^2 r}{2} + \frac{L\eta^2\gamma^2\sigma^2}{2}.$$

Summing over $t = 1, \dots, T$ and $k = 1, \dots, K$ and using $\sum_{k=1}^{K}\sum_{t=1}^{T}\mathbb{E}\big[F(W_{k,t}) - F(W_{k,t+1})\big] = F(W_{1,1}) - \mathbb{E}[F(W_{K,T+1})] \le F(W_{1,1}) - F_*$, we have

$$\bar{A} + \frac{\gamma}{2}\bar{S} \le \frac{F(W_{1,1}) - F_*}{\eta N} + 2\overline{\mathcal{E}} + \frac{L\eta r}{2} + \frac{L\eta\gamma^2\sigma^2}{2}. \tag{21}$$

Then we bound the term $\overline{\mathcal{E}}$. By applying Lemma 6, we can derive

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\tilde{E}_{k,t}\right\|_*\right] \le 2\sqrt{2r}\,\sigma\left(\sqrt{\beta} + T^{-1/2}\right) + \frac{Lr\eta}{\beta} + \frac{L\sqrt{r}\,\eta\gamma\sigma}{\sqrt{\beta}} + \frac{L\sqrt{r}\,\eta\gamma}{\beta}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|R_{k,t}\|_F\right].$$

Since $N = KT$, we can derive

$$\overline{\mathcal{E}} \le 2\sqrt{2r}\,\sigma\left(\sqrt{\beta} + T^{-1/2}\right) + \frac{Lr\eta}{\beta} + \frac{L\sqrt{r}\,\eta\gamma\sigma}{\sqrt{\beta}} + \frac{L\sqrt{r}\,\eta\gamma}{\beta}\bar{R}. \tag{22}$$

Combining Eq. (21) and Eq. (22) and using $\frac{2L\sqrt{r}\eta\gamma}{\beta}\bar{R} \le \frac{\gamma}{4}\bar{S} + \frac{4L^2 r\eta^2\gamma}{\beta^2}$, we have

$$\bar{A} + \frac{\gamma}{4}\bar{S} \le \frac{\Delta}{\eta N} + 4\sqrt{2r}\,\sigma\left(\sqrt{\beta} + T^{-1/2}\right) + \frac{2Lr\eta}{\beta} + \frac{2L\sqrt{r}\,\eta\gamma\sigma}{\sqrt{\beta}} + \frac{L\eta r}{2} + \frac{L\eta\gamma^2\sigma^2}{2} + \frac{4L^2 r\eta^2\gamma}{\beta^2}.$$

By setting $\eta = \frac{1}{2L}N^{-3/4}, \beta = N^{-1/2}, \gamma = N^{1/4}, T = K = \sqrt{N}$, we have

$$\frac{1}{N}\sum_{k=1}^{K}\sum_{t=1}^{T}\mathbb{E}\left[\left\|P_k^\top\nabla F(W_{k,t})\right\|_* + \frac{\gamma}{4}\left\|Q_k\nabla F(W_{k,t})\right\|_F^2\right] \le \left[2L\Delta + \frac{9r}{4} + \frac{\sigma^2}{4} + (8\sqrt{2}+1)\sqrt{r}\,\sigma\right]N^{-1/4}.$$

By using Eq. (20), we can obtain

$$\frac{1}{N}\sum_{k=1}^{K}\sum_{t=1}^{T}\mathbb{E}\left[\left\|P_k^\top \nabla F(W_{k,t})\right\|_* + \left\|Q_k \nabla F(W_{k,t})\right\|_F\right] \leq \left[2L\Delta + \frac{9r}{4} + \frac{\sigma^2}{4} + (8\sqrt{2}+1)\sqrt{r}\,\sigma + 1\right] N^{-1/4}.$$

According to $\|X\|_F \leq \left\|P_k^\top X\right\|_* + \|Q_k X\|_F$, we also have

$$\frac{1}{N}\sum_{k=1}^{K}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla F(W_{k,t})\right\|_F\right] \leq \left[2L\Delta + \frac{9r}{4} + \frac{\sigma^2}{4} + (8\sqrt{2}+1)\sqrt{r}\,\sigma + 1\right] N^{-1/4}.$$

C Proofs of Supporting Lemmas

C.1 Proof of Lemma 3

First, it is not hard to verify that $M_t = \beta\sum_{i=1}^{t} a^{t-i}P^\top G_i$. By summing $\omega_{i,t}$ from $i=1$ to t, we have

$$\sum_{i=1}^{t}\omega_{i,t} = \frac{\beta}{1-a^t}\sum_{j=0}^{t-1}a^j = \frac{\beta}{1-a^t}\frac{1-a^t}{1-a} = 1.$$

The weights satisfy

$$\sum_{i=1}^{t}\omega_{i,t}^2 = \frac{\beta^2}{(1-a^t)^2}\sum_{j=0}^{t-1}a^{2j} = \frac{\beta^2(1-a^{2t})}{(1-a^2)(1-a^t)^2} = \frac{\beta}{2-\beta}\frac{1+a^t}{1-a^t},$$

where the identity is due to $1-a^2 = \beta(2-\beta)$ and $1-a^{2t} = (1-a^t)(1+a^t)$.

By using $(1-\beta)^t \leq (1+\beta t)^{-1}$, we have $1-a^t \geq \frac{\beta t}{1+\beta t}$. Furthermore, by using $1+a^t \leq 2$ and $2-\beta \geq 1$, we can derive $v_t \leq 2\beta + \frac{2}{t}$, $\sqrt{v_t} \leq \sqrt{2\beta} + \sqrt{\frac{2}{t}}$.

Furthermore, we can derive

$$\sqrt{2\beta} + \frac{\sqrt{2}}{H}\sum_{t=1}^{H} t^{-1/2} \leq \sqrt{2\beta} + 2\sqrt{2}\,H^{-1/2} \leq 2\sqrt{2}\left(\sqrt{\beta} + H^{-1/2}\right),$$

where the first inequality is due to $\sum_{t=1}^{H} t^{-1/2} \leq 2\sqrt{H}$. Let $\ell = t-i$. According to the definition of $\omega_{i,t}$, we obtain

$$\sum_{i=1}^{t}\omega_{i,t}(t-i) = \frac{\beta}{1-a^t}\sum_{\ell=0}^{t-1}\ell a^\ell.$$

Moreover, we can derive

$$\sum_{\ell=0}^{t-1}\ell a^\ell = \frac{a - ta^t + (t-1)a^{t+1}}{(1-a)^2}$$

and we have

$$\sum_{i=1}^{t}\omega_{i,t}(t-i) = \frac{a - ta^t + (t-1)a^{t+1}}{\beta(1-a^t)} = \frac{a}{\beta} - \frac{ta^t}{1-a^t} \leq \frac{a}{\beta} \leq \frac{1}{\beta}.$$

Moreover, $\sum_{i=1}^{t}\omega_{i,t} = 1$. By using Jensen's inequality, we obtain

$$\sum_{i=1}^{t}\omega_{i,t}\sqrt{t-i} \leq \left(\sum_{i=1}^{t}\omega_{i,t}(t-i)\right)^{1/2} \leq \sqrt{\frac{a}{\beta}} \leq \frac{1}{\sqrt{\beta}}.$$

C.2 Proof of Lemma 4

Since $P^\top Q = P^\top - P^\top P P^\top = 0$, we can derive $\langle PZ, QG\rangle = \mathrm{tr}(Z^\top P^\top QG) = 0$. We also have $\|PZ\|_F^2 = \mathrm{tr}(Z^\top P^\top PZ) = \|Z\|_F^2$. Therefore, we can derive

$$\|PZ + \gamma QG\|_F^2 = \|Z\|_F^2 + \gamma^2 \|QG\|_F^2 + 2\gamma \langle PZ, QG\rangle = \|Z\|_F^2 + \gamma^2 \|QG\|_F^2 .$$

If $Z = \mathrm{msign}(X)$, then we have $\|Z\|_F^2 \le r$ and

$$\|PZ + \gamma QG\|_F^2 = \|Z\|_F^2 + \gamma^2 \|QG\|_F^2 + 2\gamma \langle PZ, QG\rangle \le r + \gamma^2 \|QG\|_F^2 .$$

C.3 Proof of Lemma 5

According to Assumption 1, we can derive

$$F(W_{t+1}) \le F(W_t) - \eta \langle \nabla F(W_t), PZ_t\rangle - \eta\gamma \langle \nabla F(W_t), QG_t\rangle + \frac{L\eta^2}{2} \|PZ_t + \gamma QG_t\|_F^2 .$$

According to Lemma 2, we further have

$$\langle \nabla F(W_t), PZ_t\rangle = \left\langle A_t, \mathrm{msign}(\tilde{M}_t)\right\rangle \ge \|A_t\|_* - 2\left\|\tilde{E}_t\right\|_* .$$

Since $Q^\top = Q$ and $Q^2 = Q$, we can derive $\mathbb{E}[\langle \nabla F(W_t), QG_t\rangle] = \mathbb{E}[\langle R_t, Q\nabla F(W_t)\rangle] = \mathbb{E}\left[\|R_t\|_F^2\right]$.

By using Lemma 4, we have $\mathbb{E}\left[\|PZ_t + \gamma QG_t\|_F^2\right] \le r + \gamma^2 \mathbb{E}\left[\|QG_t\|_F^2\right]$. Therefore, by using $\mathbb{E}\left[\|QG_t\|_F^2\right] \le \mathbb{E}\left[\|R_t\|_F^2\right] + \sigma^2$, we can derive

$$\mathbb{E}[F(W_{t+1})] \le \mathbb{E}[F(W_t)] - \eta\mathbb{E}[\|A_t\|_*] + 2\eta\mathbb{E}\left[\left\|\tilde{E}_t\right\|_*\right] - \eta\gamma\left(1 - \frac{L\eta\gamma}{2}\right)\mathbb{E}\left[\|R_t\|_F^2\right] + \frac{L\eta^2 r}{2} + \frac{L\eta^2\gamma^2\sigma^2}{2}.$$

Summing up from $t = 1$ to T and using $L\eta\gamma \le 1$, we conclude the proof.

C.4 Proof of Lemma 6

By using Lemma 3, $P^\top G_i = A_i + P^\top \xi_i$ and $\sum_{i=1}^t \omega_{i,t} = 1$, we have

$$\tilde{E}_t = \underbrace{\sum_{i=1}^t \omega_{i,t} P^\top \xi_i}_{\mathrm{Term}-1} + \underbrace{\sum_{i=1}^t \omega_{i,t}(A_i - A_t)}_{\mathrm{Term}-2} .$$

Then we bound the two terms separately. First we have

$$\mathbb{E}\left[\left\|\sum_{i=1}^t \omega_{i,t} P^\top \xi_i\right\|_F^2\right] \le \sum_{i=1}^t \omega_{i,t}^2 \mathbb{E}\left[\left\|P^\top \xi_i\right\|_F^2\right] \le \sigma^2 v_t.$$

By using $\|X\|_* \le \sqrt{r}\|X\|_F, \forall X \in \mathbb{R}^{r\times n}, r \le n$, we can derive

$$\mathbb{E}\left[\left\|\sum_{i=1}^t \omega_{i,t} P^\top \xi_i\right\|_*\right] \le \sigma\sqrt{r v_t}.$$

By using Lemma 3, we obtain

$$\frac{1}{T}\sum_{t=1}^T \mathbb{E}\left[\left\|\sum_{i=1}^t \omega_{i,t} P^\top \xi_i\right\|_*\right] \le 2\sqrt{2r}\sigma\left(\sqrt{\beta} + T^{-1/2}\right).$$

Under Assumption 1, we have $\|A_i - A_t\|_* \leq \sqrt{r}\, L \|W_i - W_t\|_F$. According to the update in Algorithm 3, for $1 \leq i \leq t$ we can ensure

$$\|W_i - W_t\|_F \leq \eta\sqrt{r}\,(t-i) + \eta\gamma \sum_{s=i}^{t-1} \|R_s\|_F + \eta\gamma \left\| \sum_{s=i}^{t-1} Q\xi_s \right\|_F .$$

To derive a bound on $\mathrm{Term} - 2$, we first have

$$\|\omega_{i,t}(A_i - A_t)\|_* \leq L\omega_{i,t}\eta r(t-i) + L\omega_{i,t}\sqrt{r}\eta\gamma \sum_{s=i}^{t-1} \|R_s\|_F + L\omega_{i,t}\sqrt{r}\eta\gamma \left\| \sum_{s=i}^{t-1} Q\xi_s \right\|_F .$$

Then we bound these terms one by one.

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{t}\omega_{i,t}\sum_{s=i}^{t-1}\mathbb{E}\left[\|R_s\|_F\right] &= \frac{1}{T}\sum_{t=1}^{T}\sum_{s=1}^{t-1}\left(\sum_{i=1}^{s}\omega_{i,t}\right)\mathbb{E}\left[\|R_s\|_F\right] \\
&\leq \frac{1}{T}\sum_{t=1}^{T}\sum_{s=1}^{t-1} a^{t-s}\mathbb{E}\left[\|R_s\|_F\right] \\
&= \frac{1}{T}\sum_{s=1}^{T-1}\mathbb{E}\left[\|R_s\|_F\right]\sum_{j=1}^{T-s} a^j \\
&\leq \frac{a}{\beta}\frac{1}{T}\sum_{s=1}^{T}\mathbb{E}\left[\|R_s\|_F\right] \leq \frac{\bar{R}}{\beta}.
\end{aligned}$$

We also have

$$\mathbb{E}\left[\left\|\sum_{s=i}^{t-1} Q\xi_s\right\|_F^2\right] = \sum_{s=i}^{t-1}\mathbb{E}\left[\|Q\xi_s\|_F^2\right] \leq (t-i)\sigma^2,$$

which further implies

$$\mathbb{E}\left[\left\|\sum_{s=i}^{t-1} Q\xi_s\right\|_F\right] \leq \sigma\sqrt{t-i}.$$

By using Lemma 3, we have

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{t}\omega_{i,t}\mathbb{E}\left[\left\|\sum_{s=i}^{t-1} Q\xi_s\right\|_F\right] \leq \frac{\sigma}{T}\sum_{t=1}^{T}\sum_{i=1}^{t}\omega_{i,t}\sqrt{t-i} \leq \frac{\sigma}{\sqrt{\beta}}.$$

Finally, we can obtain

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\left\|\sum_{i=1}^{t}\omega_{i,t}(A_i - A_t)\right\|_*\right] \leq \frac{Lr\eta}{\beta} + \frac{L\sqrt{r}\,\eta\gamma}{\beta}\bar{R} + \frac{L\sqrt{r}\,\eta\gamma\sigma}{\sqrt{\beta}}.$$

C.5 Proof of Lemma 7

Since $\mathbf{m}_t = (1-\beta)\mathbf{m}_{t-1} + \beta P^\top G_t$, we can verify

$$P\mathbf{m}_t = (1-\beta)P\mathbf{m}_{t-1} + \beta PP^\top G_t.$$

Then, we have

$$\begin{aligned}
&\|PP^\top\nabla F(W_t) - P\mathbf{m}_t\|_F^2 = \|PP^\top\nabla F(W_t) - (1-\beta)P\mathbf{m}_{t-1} - \beta PP^\top G_t\|_F^2 \\
&= \left\|(1-\beta)\left(PP^\top\nabla F(W_t) - P\mathbf{m}_{t-1}\right) + \beta PP^\top\left(\nabla F(W_t) - G_t\right)\right\|_F^2 .
\end{aligned}$$

By taking expectations, we have

$$\begin{aligned}
&\mathbb{E}\left[\|PP^\top \nabla F(W_t) - P\mathbf{m}_t\|_F^2\right] \\
=&(1-\beta)^2 \mathbb{E}\left[\left\|PP^\top \nabla F(W_t) - P\mathbf{m}_{t-1}\right\|_F^2\right] + \beta^2 \mathbb{E}\left[\left\|PP^\top\left(\nabla F(W_t) - G_t\right)\right\|_F^2\right] \\
\leq&(1-\beta)^2(1+\beta)\mathbb{E}\left[\left\|PP^\top \nabla F(W_{t-1}) - P\mathbf{m}_{t-1}\right\|_F^2\right] \\
&+ (1-\beta)^2\left(1+\frac{1}{\beta}\right)\mathbb{E}\left[\left\|PP^\top \nabla F(W_t) - PP^\top \nabla F(W_{t-1})\right\|_F^2\right] + \beta^2\sigma^2 \\
\leq&(1-\beta)\mathbb{E}\left[\left\|PP^\top \nabla F(W_{t-1}) - P\mathbf{m}_{t-1}\right\|_F^2\right] + \frac{1}{\beta}\mathbb{E}\left[\left\|PP^\top \nabla F(W_t) - PP^\top \nabla F(W_{t-1})\right\|_F^2\right] + \beta^2\sigma^2,
\end{aligned} \tag{23}$$

where the first inequality is due to the weighted Young inequality $\|x+y\|_F^2 \leq (1+\theta)\|x\|_F^2 + (1+1/\theta)\|y\|_F^2$. To bound the second term on the right-hand side, we use Assumption 1 and obtain

$$\begin{aligned}
&\left\|PP^\top \nabla F(W_t) - PP^\top \nabla F(W_{t-1})\right\|_F^2 \leq \|\nabla F(W_t) - \nabla F(W_{t-1})\|_F^2 \\
\leq& L^2 \|W_t - W_{t-1}\|_F^2 = L^2 \left\|\eta P\mathbf{m}_{t-1} + \eta\gamma(I - PP^\top)G_{t-1}\right\|_F^2 \\
=&\eta^2 L^2 \|P\mathbf{m}_{t-1}\|_F^2 + \eta^2\gamma^2 L^2 \left\|(I-PP^\top)G_{t-1}\right\|_F^2,
\end{aligned}$$

where the last term is the undesirable contribution from the SGD correction, which causes this direct decomposition argument to break down. To make this explicit, we substitute the above bound into Eq. (23) and obtain

$$\begin{aligned}
&\mathbb{E}\left[\|PP^\top \nabla F(W_t) - P\mathbf{m}_t\|_F^2\right] \\
\leq&(1-\beta)\mathbb{E}\left[\left\|PP^\top \nabla F(W_{t-1}) - P\mathbf{m}_{t-1}\right\|_F^2\right] + \frac{\eta^2 L^2}{\beta}\mathbb{E}\left[\|P\mathbf{m}_{t-1}\|_F^2\right] \\
&+ \frac{\eta^2\gamma^2 L^2}{\beta}\mathbb{E}\left[\left\|(I-PP^\top)G_{t-1}\right\|_F^2\right] + \beta^2\sigma^2.
\end{aligned}$$

Summing over $t = 2, \ldots, T+1$, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=2}^{T+1}\|PP^\top \nabla F(W_t) - P\mathbf{m}_t\|_F^2\right] \\
\leq&(1-\beta)\mathbb{E}\left[\sum_{t=2}^{T+1}\left\|PP^\top \nabla F(W_{t-1}) - P\mathbf{m}_{t-1}\right\|_F^2\right] + \frac{\eta^2 L^2}{\beta}\mathbb{E}\left[\sum_{t=2}^{T+1}\|P\mathbf{m}_{t-1}\|_F^2\right] \\
&+ \frac{\eta^2\gamma^2 L^2}{\beta}\mathbb{E}\left[\sum_{t=2}^{T+1}\left\|(I-PP^\top)G_{t-1}\right\|_F^2\right] + \beta^2\sigma^2 T \\
=&(1-\beta)\mathbb{E}\left[\sum_{t=1}^{T}\left\|PP^\top \nabla F(W_t) - P\mathbf{m}_t\right\|_F^2\right] + \frac{\eta^2 L^2}{\beta}\mathbb{E}\left[\sum_{t=1}^{T}\|P\mathbf{m}_t\|_F^2\right] \\
&+ \frac{\eta^2\gamma^2 L^2}{\beta}\mathbb{E}\left[\sum_{t=1}^{T}\left\|(I-PP^\top)G_t\right\|_F^2\right] + \beta^2\sigma^2 T.
\end{aligned}$$

Rearranging the above inequality, we obtain Eq. (26).

D Limitations of a Direct Decomposition Analysis

This section explains why a seemingly simpler orthogonal-decomposition proof does not yield the desired rate. According to Assumption 1, we have

$$\begin{aligned}
F(W_{t+1}) \leq& F(W_t) + \langle \nabla F(W_t), W_{t+1} - W_t\rangle + \frac{L}{2}\|W_{t+1} - W_t\|_F^2 \\
=& F(W_t) - \langle \nabla F(W_t), \eta P\mathbf{m}_t + \eta\gamma(I-PP^\top)G_t\rangle + \frac{L}{2}\|\eta P\mathbf{m}_t + \eta\gamma(I-PP^\top)G_t\|_F^2.
\end{aligned}$$

Since $P\mathbf{m}_t$ and $(I - PP^\top)G_t$ are orthogonal and the quadratic term expands without a cross term, we can derive

$$\begin{aligned} F(W_{t+1}) \leq & F(W_t) \underbrace{-\eta\langle \nabla F(W_t), P\mathbf{m}_t\rangle + \frac{\eta^2 L}{2}\|P\mathbf{m}_t\|_F^2}_{\mathrm{Term}-1(t)} \\ & \underbrace{-\eta\gamma\langle \nabla F(W_t), (I-PP^\top)G_t\rangle + \frac{\eta^2\gamma^2 L}{2}\|(I-PP^\top)G_t\|_F^2}_{\mathrm{Term}-2(t)}. \end{aligned} \tag{24}$$

Here, $\mathrm{Term}-1(t)$ corresponds to the momentum-SGD update in $\mathrm{span}(P)$, whereas $\mathrm{Term}-2(t)$ corresponds to the residual SGD correction in $\mathrm{span}(P)^\perp$. A natural approach is to analyze $\mathrm{Term}-1(t)$ using a standard momentum-tracking argument and $\mathrm{Term}-2(t)$ using a standard SGD descent analysis, and then combine the resulting bounds. We show below that this direct decomposition fails to establish the desired convergence rate.

To control $\mathrm{Term}-1(t)$, we follow existing work (Jiang et al., 2023) and have

$$\begin{aligned} \mathrm{Term}-1(t) = & -\eta\langle \nabla F(W_t), P\mathbf{m}_t\rangle + \frac{\eta}{2}\|PP^\top\nabla F(W_t)\|_F^2 + \frac{\eta}{2}\|P\mathbf{m}_t\|_F^2 \\ & - \frac{\eta}{2}\|PP^\top\nabla F(W_t)\|_F^2 - \frac{\eta}{2}\|P\mathbf{m}_t\|_F^2 + \frac{\eta^2 L}{2}\|P\mathbf{m}_t\|_F^2 \\ \leq & \frac{\eta}{2}\|PP^\top\nabla F(W_t) - P\mathbf{m}_t\|_F^2 - \frac{\eta}{2}\|PP^\top\nabla F(W_t)\|_F^2 - \frac{\eta}{4}\|P\mathbf{m}_t\|_F^2, \end{aligned}$$

where the inequality is due to $\eta \leq 1/(2L)$. Summing the above inequality over $t = 1, \ldots, T$, we have

$$\sum_{t=1}^T \mathrm{Term}-1(t) \leq \frac{\eta}{2}\sum_{t=1}^T \|PP^\top\nabla F(W_t) - P\mathbf{m}_t\|_F^2 - \frac{\eta}{2}\sum_{t=1}^T \|PP^\top\nabla F(W_t)\|_F^2 - \frac{\eta}{4}\sum_{t=1}^T \|P\mathbf{m}_t\|_F^2. \tag{25}$$

It remains to bound the first term, which quantifies the tracking error between $P\mathbf{m}_t$ and $PP^\top\nabla F(W_t)$. To this end, we have the following lemma.

Lemma 7 *For the momentum estimator in Algorithm 1, we have*

$$\begin{aligned} & \mathbb{E}\left[\sum_{t=1}^T \|PP^\top\nabla F(W_t) - P\mathbf{m}_t\|_F^2\right] \\ \leq & \frac{\mathbb{E}\left[\|PP^\top\nabla F(W_1) - P\mathbf{m}_1\|_F^2 - \|PP^\top\nabla F(W_{T+1}) - P\mathbf{m}_{T+1}\|_F^2\right]}{\beta} + \frac{\eta^2 L^2}{\beta^2}\mathbb{E}\left[\sum_{t=1}^T \|P\mathbf{m}_t\|_F^2\right] \\ & + \frac{\eta^2\gamma^2 L^2}{\beta^2}\mathbb{E}\left[\sum_{t=1}^T \left\|(I-PP^\top)G_t\right\|_F^2\right] + \beta\sigma^2 T. \end{aligned} \tag{26}$$

Combining Eq. (26) and Eq. (25), and using $\eta \leq \beta/(2L)$, we have

$$\begin{aligned} \mathbb{E}\left[\sum_{t=1}^T \mathrm{Term}-1(t)\right] \leq & \frac{\eta}{2\beta}\left(\mathbb{E}\left[\|PP^\top\nabla F(W_1) - P\mathbf{m}_1\|_F^2 - \|PP^\top\nabla F(W_{T+1}) - P\mathbf{m}_{T+1}\|_F^2\right]\right) \\ & + \frac{\eta\beta\sigma^2 T}{2} - \frac{\eta}{2}\mathbb{E}\left[\sum_{t=1}^T \|PP^\top\nabla F(W_t)\|_F^2\right] + \frac{\eta^2\gamma^2 L}{4\beta}\mathbb{E}\left[\sum_{t=1}^T \left\|(I-PP^\top)G_t\right\|_F^2\right]. \end{aligned} \tag{27}$$

The last term in Eq. (27) is where this route fails. Lemma 7 must control the change of $PP^\top\nabla F(W_t)$ between consecutive iterates. Smoothness bounds this change by $L\|W_t - W_{t-1}\|_F$

and the step contains the residual correction $\eta\gamma(I - PP^\top)G_{t-1}$, so the residual enters the tracking-error bound with coefficient $\eta^2\gamma^2 L/(4\beta)$. The residual descent in $\sum_t \mathrm{Term} - 2(t)$ has coefficient $\eta\gamma/2$, so the two differ by a factor $\eta\gamma L/(2\beta)$. The remaining terms in Eq. (27) force $\beta = \Theta(1/\sqrt{T})$ and $\eta = \Theta(1/\sqrt{T})$ and for constant γ this factor is then $\Theta(1)$.

Next, we bound $\mathrm{Term} - 2(t)$ based on a standard analysis:

$$
\begin{aligned}
&\mathbb{E}\left[\mathrm{Term} - 2(t)\right] \\
=&\mathbb{E}\left[-\eta\gamma\langle\nabla F(W_t), (I - PP^\top)G_t\rangle + \frac{\eta^2\gamma^2 L}{2}\|(I - PP^\top)G_t\|_F^2\right] \\
=&-\eta\gamma\mathbb{E}\left[\|(I - PP^\top)\nabla F(W_t)\|_F^2\right] + \frac{\eta^2\gamma^2 L}{2}\mathbb{E}\left[\|(I - PP^\top)(G_t - \nabla F(W_t)) + (I - PP^\top)\nabla F(W_t)\|_F^2\right] \\
\le&-\eta\gamma\mathbb{E}\left[\|(I - PP^\top)\nabla F(W_t)\|_F^2\right] + \frac{\eta^2\gamma^2 L}{2}\mathbb{E}\left[\|(I - PP^\top)\nabla F(W_t)\|_F^2\right] + \frac{\eta^2\gamma^2 L\sigma^2}{2} \\
\le&-\frac{\eta\gamma}{2}\mathbb{E}\left[\|(I - PP^\top)\nabla F(W_t)\|_F^2\right] + \frac{\eta^2\gamma^2 L\sigma^2}{2},
\end{aligned}
$$

where the last inequality is due to $\eta\gamma \le 1/L$. Summing the above inequality over $t = 1, \ldots, T$, we have

$$
\mathbb{E}\left[\sum_{t=1}^{T}\mathrm{Term} - 2(t)\right] \le -\frac{\eta\gamma}{2}\mathbb{E}\left[\sum_{t=1}^{T}\|(I - PP^\top)\nabla F(W_t)\|_F^2\right] + \frac{\eta^2\gamma^2 L\sigma^2 T}{2}. \tag{28}
$$

Summing (24) over all rounds, taking expectations, and setting $\gamma \le 1$, we have

$$
\begin{aligned}
\mathbb{E}\left[F(W_{T+1}) - F(W_1)\right] \le &-\frac{\eta\gamma}{2}\mathbb{E}\left[\sum_{t=1}^{T}\|\nabla F(W_t)\|_F^2\right] + \frac{\eta^2\gamma^2 L\sigma^2 T}{2} + \frac{\eta\beta\sigma^2 T}{2} \\
&+ \frac{\eta}{2\beta}\mathbb{E}\Big[\|PP^\top\nabla F(W_1) - P\mathbf{m}_1\|_F^2 - \|PP^\top\nabla F(W_{T+1}) - P\mathbf{m}_{T+1}\|_F^2\Big] \\
&+ \frac{\eta^2\gamma^2 L}{4\beta}\mathbb{E}\left[\sum_{t=1}^{T}\left\|(I - PP^\top)G_t\right\|_F^2\right],
\end{aligned}
$$

where the inequality is due to Eq. (27) and Eq. (28). Finally, we have

$$
\begin{aligned}
&\frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\|\nabla F(W_t)\|_F^2\right] \\
\le&\frac{2}{\eta\gamma T}\mathbb{E}\left[F(W_1) - F(W_{T+1})\right] \\
&+ \frac{1}{\beta\gamma T}\left(\mathbb{E}\left[\|PP^\top\nabla F(W_1) - P\mathbf{m}_1\|_F^2 - \|PP^\top\nabla F(W_{T+1}) - P\mathbf{m}_{T+1}\|_F^2\right]\right) \\
&+ \left(\eta\gamma L + \frac{\beta}{\gamma}\right)\sigma^2 + \frac{\eta\gamma L}{2\beta T}\mathbb{E}\left[\sum_{t=1}^{T}\left\|(I - PP^\top)G_t\right\|_F^2\right].
\end{aligned} \tag{29}
$$

Without the last term in Eq. (29), the choices $\gamma = \Theta(1)$ and $\eta, \beta = \Theta(T^{-1/2})$ would yield an $O(T^{-1/2})$ bound on the average squared gradient norm. However, under this scaling the coefficient $\eta\gamma L/(2\beta)$ remains $\Theta(1)$. Although the true-gradient component of the residual second moment can be absorbed into the descent term, bounding its noise component by σ^2 leaves a variance contribution that does not vanish with T. Consequently, this tracking bound does not establish the desired $O(T^{-1/2})$ rate under stochastic gradients.

E Experiments

In this section, we assess the performance of the proposed methods via numerical experiments.

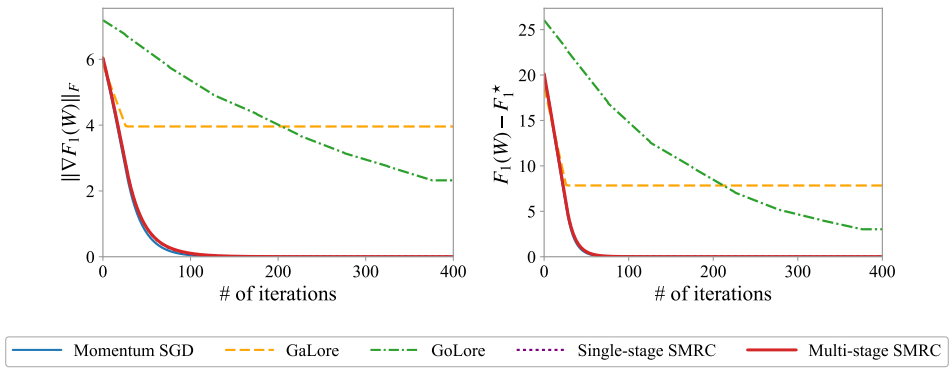

Figure 1: Effect of residual correction.

E.1 Synthetic Experiments

We conduct two synthetic experiments and compare our methods with Momentum SGD, GaLore (Zhao et al., 2024) and GoLore (He et al., 2025).

E.1.1 Effect of Residual Correction

Setup. We conduct this experiment using a stochastic quadratic problem adapted from the counterexample of He et al. (2025):

$$F_1(W) = \frac{1}{2}\|AW\|_F^2 + \langle B, W\rangle, \qquad G_t = \nabla F_1(W_t) + \xi_t, \tag{30}$$

where $W \in \mathbb{R}^{m\times n}$ and ξ_t is gradient noise with $\mathbb{E}[\xi_t] = 0$. We set $m = 30, n = 8$ and the projection rank $r = 2$. Because the number of rows exceeds the number of columns, all low-rank methods use a right projection basis $P_t \in \mathbb{R}^{8\times 2}$. For GaLore, GoLore and multi-stage SMRC, we set the refresh interval to 50. We train all methods for 400 iterations and grid search the parameters $\eta \in \{0.05, 0.1, 0.2, 0.5, 0.8, 0.9\}$ and $\beta \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. We set $\gamma = 0.8$ for SMRC. We set

$$U_s = \begin{bmatrix} I_2 \\ \mathbf{0}_{28\times 2} \end{bmatrix} \in \mathbb{R}^{30\times 2}, \quad U_n = \begin{bmatrix} \mathbf{0}_{28\times 2} \\ I_2 \end{bmatrix} \in \mathbb{R}^{30\times 2},$$

which satisfy $U_s^\top U_s = I_2, U_n^\top U_n = I_2, U_s^\top U_n = \mathbf{0}$. We further define

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}.$$

The quadratic objective is specified by $A = U_s^\top, B = \frac{1}{2}U_s H$, where $A \in \mathbb{R}^{2\times 30}$ has orthonormal rows. The parameter matrix is initialized as $W_0 = U_s \operatorname{diag}(1.8, 0.9)H$. We define $V_s = \frac{1}{\sqrt{8}}H^\top$ and compute a complete QR factorization of V_s and form $V_n \in \mathbb{R}^{8\times 2}$ from the third and fourth columns of the orthogonal matrix. We have $V_n^\top V_n = I_2, V_n^\top V_s = \mathbf{0}$. At iteration t, the stochastic gradient is $G_t = \nabla F_1(W_t) + \xi_t$, where $\xi_t = 4 s_t U_n V_n^\top, s_t \overset{\text{i.i.d.}}{\sim} \text{Rademacher}(\{-1, +1\})$.

Results. As shown in Figure 1, GaLore fails to converge when its selected subspace becomes dominated by gradient noise, leaving informative gradient directions outside the subspace. By incorporating the discarded gradient components through residual correction, SMRC restores convergence and achieves performance comparable to that of Momentum SGD.

E.1.2 Benefit of Multi-Stage Subspace Refreshing

Setup. We consider the following problem

$$F_2(W) = \frac{\lambda_1}{2}\|U_1^\top W\|_F^2 + \frac{\lambda_2}{2}\|U_2^\top W\|_F^2 + \frac{\mu}{2}\|U_\perp^\top W\|_F^2, \qquad G_t = \nabla F_2(W_t) + \xi_t, \tag{31}$$

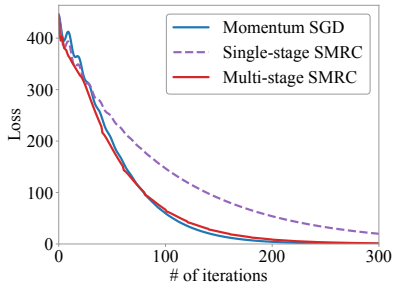

Figure 2: Results of multi-stage refreshing.

where $W \in \mathbb{R}^{30\times 32}$. Let $e_i \in \mathbb{R}^{30}$ be the i-th standard unit vector. We set $U_1 = [e_1, e_2]$, $U_2 = [e_3, e_4]$, $U_\perp = [e_5, \ldots, e_{30}]$, $(\lambda_1, \lambda_2, \mu) = (4, 0.02, 0.01)$ and $W_0 = 1.2U_1H_1 + 50U_2H_2$, where

$$H_1 = \begin{bmatrix} \mathbf{1}_8^\top & \mathbf{0}_8^\top & \mathbf{0}_8^\top & \mathbf{0}_8^\top \\ \mathbf{0}_8^\top & \mathbf{1}_8^\top & \mathbf{0}_8^\top & \mathbf{0}_8^\top \end{bmatrix}, \qquad H_2 = \begin{bmatrix} \mathbf{0}_8^\top & \mathbf{0}_8^\top & \mathbf{1}_8^\top & \mathbf{0}_8^\top \\ \mathbf{0}_8^\top & \mathbf{0}_8^\top & \mathbf{0}_8^\top & \mathbf{1}_8^\top \end{bmatrix}.$$

We use Gaussian noise $\xi_t = 0.01Z_t$, where $(Z_t)_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$. We set the projection rank to $r = 2$ and train for 300 iterations. Single-stage SMRC keeps the subspace fixed, whereas multi-stage SMRC refreshes it every 20 iterations. At each stage transition, multi-stage SMRC explicitly applies the boundary correction in Algorithm 2. For all methods, we search $\eta, \beta \in \{0.02, 0.05, 0.1, 0.3, 0.5\}$. For SMRC variants, we additionally search over $\gamma \in \{0.02, 0.05, 0.1, 0.3, 0.5, 0.7\}$.

Results. Figure 2 shows that single-stage SMRC continues to make progress through residual correction, but its convergence slows as the dominant gradient directions evolve during optimization. In contrast, multi-stage SMRC periodically refreshes the projection basis to track the evolving dominant directions, resulting in faster convergence and performance comparable to that of full-space Momentum SGD.

E.2 Language Model Experiments

To verify the empirical effectiveness of the proposed algorithm in practice, we compare our PMRC with previous methods in LLM pretraining and finetuning settings.

E.2.1 A practical implementation of PMRC

We present a practical variant of Algorithm 4 with the following two modifications.

First, although the residual correction is explicitly scaled by γ, a single global coefficient cannot consistently align its magnitude with the projected Muon direction across parameter blocks and optimization steps. In particular, during early training, the initial subspace may leave a relatively large fraction of the gradient in the orthogonal complement, causing $\gamma\|(I - P_kP_k^\top)G_{k,t}\|_F$ to substantially exceed $\|P_kZ_{k,t}\|_F$. We observed that this scale imbalance coincides with pronounced loss fluctuations during the early stage of training. We introduce a residual cap that bounds the residual correction with respect to the projected Muon direction, while preserving the original coefficient γ whenever the residual update is already below the prescribed threshold.

Define $R_{k,t} = (I - P_kP_k^\top)G_{k,t}$. We modify the update rule as follows:

$$W_{k,t+1} = W_{k,t} - 0.2\sqrt{\max\{m,n\}}\,\eta\,(P_kZ_{k,t} + \widehat{\gamma}_{k,t}R_{k,t}) - \eta\lambda W_{k,t}, \tag{32}$$

where

$$\widehat{\gamma}_{k,t} = \min\left\{\gamma, \rho\sqrt{\frac{\|P_kZ_{k,t}\|_F^2}{\max\{\|R_{k,t}\|_F^2, \epsilon\}}}\right\}.$$

Table 3: Per-matrix projector latency on the LLaMA-350M matrix shapes on a single NVIDIA RTX PRO 6000 GPU.

Matrix Shape	Exact SVD (ms)	Ours (ms)
1024×1024	58.01	3.21
2736×1024	57.75	2.98
1024×2736	57.36	2.98

Table 4: Architecture configurations of the LLaMA-style models.

Model Size	Hidden Size	Layers	Heads	FFN Size	Vocabulary Size	Max Sequence Length
60M	512	8	8	1376	32000	1024
150M	768	14	12	2048	32000	2048
350M	1024	24	16	2736	32000	2048
1.3B	2048	24	32	5461	32000	1024

Following the standard practical implementation of Muon, we retain the shape-dependent learning-rate scaling factor $0.2\sqrt{\max\{m,n\}}$ and weight decay with coefficient λ. These two components are inherited from Liu et al. (2025). We set $\rho = 4$ and $\epsilon = 10^{-30}$ in our experiments.

Second, periodic SVD computations in GaLore introduce time overhead when the projection basis is refreshed, particularly for high-dimensional matrices. To improve computational efficiency, we replace the exact SVD with a warm-started subspace approximation adapted from ARO (Gong et al., 2026). Specifically, given the matrix $G \in \mathbb{R}^{m\times n}$ and the previous basis $P^{\text{prev}} \in \mathbb{R}^{m\times r}$, we update the basis by

$$P = \mathrm{QR}\Big(G\, f_{\mathrm{RN}}\big((P^{\text{prev}})^\top G\big)^\top\Big),$$

where $f_{\mathrm{RN}}(\cdot)$ performs row normalization and $\mathrm{QR}(\cdot)$ returns an orthonormal basis. The corresponding singular values are estimated from the row norms of $P^\top G$. In our implementation, the QR operation is computed with SCQR (Fukaya et al., 2020). To quantify the practical benefit, we measure the per-matrix latency of a basis refresh on the LLaMA-350M shapes. As shown in Table 3, the approximate refresh consistently takes around 3 milliseconds, whereas the exact SVD used by GaLore requires nearly 58 milliseconds.

E.2.2 LLM Pretraining Tasks

We adopt the LLaMA architecture and study four model scales with 60M, 150M, 350M and 1.3B parameters, whose architectural configurations are summarized in Table 4. We pretrain all models on the C4 dataset (Raffel et al., 2020). We implement the experiments based on the code from Glentis et al. (2025) [1] and calculate $\mathrm{msign}(\cdot)$ using Newton–Schulz iterations. Our implementations of GaLore[2], GoLore[3] and Fira[4] are based on their publicly released code. For AdamW-based methods, we set $\beta_1 = 0.9$ and $\beta_2 = 0.999$. For Muon-based methods, we set the momentum coefficient to $1-\beta = 0.95$ and enable Nesterov momentum. We compute the matrix sign using 5 Newton–Schulz iterations and optimize the one-dimensional and embedding-layer parameters with AdamW.

We then evaluate the models on the following benchmarks: ARC-Easy, ARC-Challenge (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019) and WinoGrande (Sakaguchi et al., 2019).

Each hyperparameter either follows the recommendation of the original paper or is selected by grid search. For each model and optimizer, we select the hyperparameter configuration that achieves the highest average accuracy across the seven downstream benchmarks, with the chosen values summarized in Table 7. We report the memory cost of all methods in Table 6, measured with

[1] https://github.com/OptimAI-Lab/Minimalist_LLM_Pretraining
[2] https://github.com/jiaweizzhao/galore
[3] https://github.com/pkumelon/golore
[4] https://github.com/xichen-fy/Fira

Table 5: Training configurations of the LLaMA-style models.

Hyperparameter	60M	150M	350M	1.3B
Global Batch Size	512	512	640	1280
Device Micro-batch Size	256	256	160	160
Sequence Length	512	512	1024	1024
Tokens per Step	262144	262144	655360	1310720
Training Steps	4431	9900	11300	20480
Total Training Tokens	1.162B	2.595B	7.406B	26.84B
Learning-rate Schedule	cosine	cosine	cosine	cosine
Gradient Clipping	1.0	1.0	1.0	1.0

Table 6: Peak allocated GPU memory (GiB) during 1.3B pretraining.

Model	**AdamW**	**Muon**	**GaLore**	**GoLore**	**Fira**	**GUM**	**PMRC**
1.3B	57.77	55.52	54.00	54.00	54.00	53.85	**53.44**

`torch.compile` disabled. As shown in Table 8, PMRC achieves the highest average accuracy among all memory-efficient methods across all model scales.

E.2.3 LLM Finetuning Tasks

In this section, we provide the omitted details of the LLM supervised finetuning (SFT) experiments. For the instruction-following task, we train on a single NVIDIA RTX PRO 6000 GPU and adopt a device micro-batch size of 16 with 8 gradient accumulation steps for Muon, GaLore, GoLore, GUM, and PMRC, and a device micro-batch size of 8 with 16 gradient accumulation steps for AdamW, so that the global batch size is 128 in both cases. For the mathematical reasoning task, we train on 4 NVIDIA H20 GPUs and adopt a device micro-batch size of 4 with 8 gradient accumulation steps for the same methods and a device micro-batch size of 2 with 16 gradient accumulation steps for AdamW, so that the global batch size is again 128. For AdamW-based methods, we use the standard coefficients $\beta_1 = 0.9$ and $\beta_2 = 0.999$. For Muon-based methods, the momentum coefficient is set to $1 - \beta = 0.95$ with Nesterov momentum enabled. The matrix sign is approximated using 5 Newton–Schulz iterations, while one-dimensional parameters and embedding layers are optimized with AdamW.

Hyperparameter Tuning. For each model and optimization method we select the learning rate by a grid search over $\{1 \times 10^{-5}, 2 \times 10^{-5}, 4 \times 10^{-5}, 8 \times 10^{-5}\}$ for the instruction-following task and $\{8 \times 10^{-6}, 1 \times 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}, 7 \times 10^{-5}\}$ for the mathematical-reasoning task. PMRC uses a residual cap at a ratio of 4 and we grid search $\gamma \in \{2000, 4000, 8000, 16000\}$. We select the parameters that achieve the highest accuracy on benchmarks.

Additional results. We present the prompt-level and instruction-level strict accuracy on the IFEval benchmark in Table 9. For completeness, we include the results of GUM with rank $r = 256$, in addition to the memory-matched setting with $r = 128$ used in the main experiments. PMRC outperforms GUM at both ranks on all three models in both metrics. We also report the peak allocated GPU memory of all methods in Table 10, measured on a single NVIDIA H20 GPU with `torch.compile` disabled. PMRC requires the lowest peak memory across all models.

E.2.4 Ablation Studies

For all ablations, we pretrain LLaMA-150M on the C4 dataset for 9900 steps with a sequence length of 512 and a global batch size of 512. We refresh the projection basis every 100 steps and, unless otherwise stated, set the residual cap to $\rho = 4$.

Effect of the subspace rank. We vary the projection rank over $r \in \{64, 128, 256, 512\}$ with the residual weight fixed at $\gamma = 8192$ and all other settings unchanged. Increasing r expands the subspace in which momentum is maintained and the matrix-sign transformation is applied. As shown in Figure 3, a larger rank reduces the validation loss, but also increases the memory required

Table 7: Hyperparameter settings for different optimizers and model sizes.

Model	AdamW	Muon	GaLore	GoLore	Fira	GUM	PMRC	
	lr	lr	lr	lr	lr	lr	lr	γ
LLaMA-60M	3×10^{-3}	3×10^{-3}	3×10^{-3}	5×10^{-3}	2×10^{-3}	3×10^{-3}	2×10^{-3}	8192
LLaMA-150M	2×10^{-3}	3×10^{-3}	5×10^{-3}	5×10^{-3}	5×10^{-3}	5×10^{-3}	2×10^{-3}	16384
LLaMA-350M	1×10^{-3}	2×10^{-3}	5×10^{-3}	5×10^{-3}	5×10^{-3}	3×10^{-3}	2×10^{-3}	16384
LLaMA-1.3B	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	5×10^{-3}	16384

Table 8: Zero-shot accuracy (%) on seven benchmarks. **Bold** and underlined values indicate the best and second-best average accuracies among memory-efficient methods, respectively. Averages are computed from unrounded scores.

Model	Optimizer	ARC-E	ARC-C	OBQA	HellaSwag	PIQA	SIQA	WinoGrande	Avg.
LLaMA -60M	AdamW	33.67	17.92	12.80	27.15	59.74	35.98	51.07	34.05
	Muon	35.77	17.83	13.20	27.46	60.66	35.82	49.80	34.37
	GaLore	34.39	18.60	14.40	26.94	58.87	34.75	52.33	<u>34.32</u>
	GoLore	32.79	17.75	12.20	27.03	57.83	37.10	52.01	33.82
	Fira	34.93	18.86	12.20	26.98	59.19	35.47	50.75	34.05
	GUM	35.02	18.34	13.40	27.04	58.27	35.98	51.62	34.23
	PMRC	34.89	18.43	13.40	27.32	59.41	36.69	52.41	**34.65**
LLaMA -150M	AdamW	38.59	17.49	14.60	28.39	61.64	35.67	50.83	35.32
	Muon	40.03	17.66	16.40	28.64	62.73	36.80	50.43	36.10
	GaLore	38.22	17.58	13.60	27.99	60.61	37.10	50.51	35.09
	GoLore	38.13	16.38	15.00	27.76	60.88	36.34	50.99	35.07
	Fira	38.59	18.17	13.40	28.15	61.64	35.88	50.83	35.24
	GUM	38.09	18.34	14.60	27.89	60.77	37.15	50.75	<u>35.37</u>
	PMRC	38.76	18.77	15.80	28.81	61.21	36.80	52.17	**36.05**
LLaMA -350M	AdamW	42.68	18.00	16.80	30.14	64.96	37.15	52.80	37.51
	Muon	44.19	19.97	18.80	32.08	66.21	38.02	54.70	39.14
	GaLore	40.03	19.11	15.20	29.13	62.95	36.85	51.62	36.41
	GoLore	39.86	16.55	15.00	29.20	63.66	37.05	50.20	35.93
	Fira	41.50	17.58	16.20	29.94	64.04	36.85	51.54	36.81
	GUM	42.13	19.54	15.60	30.15	65.78	37.31	50.83	<u>37.33</u>
	PMRC	44.11	20.14	18.40	31.25	65.67	38.54	50.99	**38.44**
LLaMA -1.3B	AdamW	46.68	21.25	18.40	34.79	68.28	39.15	51.62	40.02
	Muon	51.60	21.76	19.60	39.14	71.00	40.53	51.14	42.11
	GaLore	42.00	19.45	15.60	29.93	64.64	37.41	50.04	37.01
	GoLore	38.80	17.66	13.40	28.90	62.84	34.80	51.30	35.39
	Fira	45.03	18.77	16.20	32.10	67.25	37.56	51.85	38.40
	GUM	46.97	20.14	19.00	33.62	67.25	39.71	52.25	<u>39.85</u>
	PMRC	48.86	20.65	19.60	36.01	69.48	39.76	51.07	**40.78**

to store the projection basis and the projected momentum. For a parameter matrix of size $m \times n$, this memory cost scales as $O((m+n)r)$. These results illustrate the trade-off between validation performance and optimizer-state memory when choosing the subspace rank.

Effect of the residual weight. We fix the rank at $r = 256$ and vary the residual weight over $\gamma \in \{0, 1024, 2048, 4096, 8192\}$ with all other settings unchanged. Since the residual cap remains active, γ serves as an upper bound on the effective weight applied to each parameter matrix rather than its exact value. As shown in Figure 4, the validation loss decreases monotonically as γ increases over the tested range, with diminishing improvements at larger values. In particular, $\gamma = 0$ removes the residual correction and yields the highest validation loss. This comparison supports the benefit of incorporating gradient information from the orthogonal complement in LLM pretraining.

Table 9: Accuracy (%) on the IFEval (instruction-following) benchmark. **Bold** and underlined values indicate the best and second-best results among memory-efficient methods, respectively.

		Prompt-level Strict Accuracy			Instruction-level Strict Accuracy		
Memory-Efficient	**Optimizer**	**Qwen3-8B**	**LLaMA-3.1-8B**	**OLMo-2-1124-7B**	**Qwen3-8B**	**LLaMA-3.1-8B**	**OLMo-2-1124-7B**
×	AdamW	56.38	39.19	26.80	65.83	50.12	38.97
×	Muon	52.68	34.38	24.03	62.11	46.16	36.45
✓	GaLore	57.49	36.60	26.62	66.67	47.24	38.13
✓	GoLore	58.96	32.90	24.03	68.35	47.96	35.73
✓	GUM ($r = 128$)	51.76	33.46	24.95	62.47	43.65	36.21
✓	GUM ($r = 256$)	55.45	35.30	26.25	65.35	46.76	36.69
✓	**PMRC**	**59.89**	**37.52**	**27.36**	**69.06**	**50.24**	**38.97**

Table 10: Peak allocated GPU memory (GiB) during full-parameter SFT. We use a micro-batch size of 16 with 8 gradient accumulation steps for IFEval and a micro-batch size of 4 with 32 gradient accumulation steps for GSM8K. All memory measurements are performed on a single H20 GPU.

Task	**Model**	**AdamW**	**Muon**	**GaLore**	**GoLore**	**GUM**	**PMRC**
IFEval	Qwen3-8B	89.16	76.23	67.71	67.71	66.01	**65.88**
	LLaMA-3.1-8B	78.32	65.32	56.63	56.63	54.90	**54.81**
	OLMo-2-1124-7B	69.74	57.43	49.23	49.23	47.79	**47.75**
GSM8K	Qwen3-8B	79.46	66.52	58.02	58.02	57.00	**56.18**
	LLaMA-3.1-8B	74.98	61.98	53.30	53.30	52.35	**51.48**
	OLMo-2-1124-7B	66.77	54.71	46.54	46.54	45.80	**45.03**

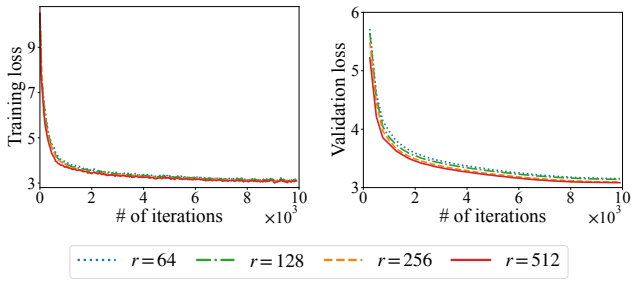

Figure 3: Effect of the subspace rank r on PMRC.

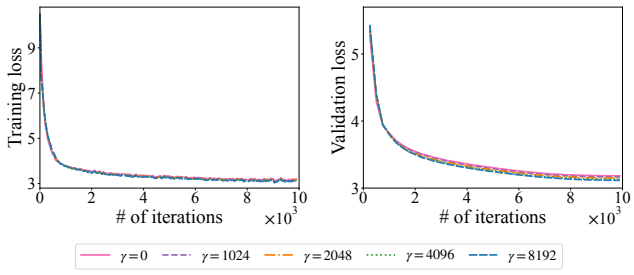

Figure 4: Effect of the residual weight γ on PMRC.

Effect of the residual cap. We finally examine the role of the residual cap in stabilizing training. Under the practical update in Eq. (32), the effective residual weight is

$$\widehat{\gamma}_{k,t} = \min\left\{\gamma, \rho\sqrt{\frac{\|P_k Z_{k,t}\|_F^2}{\max\{\|R_{k,t}\|_F^2, \epsilon\}}}\right\}.$$

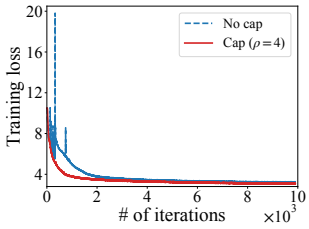

Figure 5: Effect of the residual cap on PMRC.

By construction, this rule ensures $\|\widehat{\gamma}_{k,t} R_{k,t}\|_F \leq \rho \|P_k Z_{k,t}\|_F$. We compare the default choice $\rho = 4$ with an uncapped run, fixing $r = 256$ and $\gamma = 8192$ in both runs. Without the cap, the full residual gradient is multiplied by γ at every step. As shown in Figure 5, the uncapped run becomes unstable during early training. This observation supports the use of the residual cap to control excessively large residual updates, while retaining the original coefficient γ whenever the correction is below the threshold.