
SUMO: Subspace-Aware Moment-Orthogonalization for Accelerating Memory-Efficient LLM Training

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Low-rank gradient-based optimization methods have significantly improved mem-
2 ory efficiency during the training of large language models (LLMs), enabling
3 operations within constrained hardware without sacrificing performance. However,
4 these methods primarily emphasize memory savings, often overlooking potential
5 acceleration in convergence due to their reliance on standard isotropic steepest
6 descent techniques, which can perform suboptimally in the highly anisotropic
7 landscapes typical of deep networks, particularly LLMs. In this paper, we propose
8 SUMO (Subspace-Aware Moment-Orthogonalization), an optimizer that employs
9 exact singular value decomposition (SVD) for moment orthogonalization within
10 a dynamically adapted low-dimensional subspace, enabling norm-inducing steep-
11 est descent optimization steps. By explicitly aligning optimization steps with
12 the spectral characteristics of the loss landscape, SUMO effectively mitigates ap-
13 proximation errors associated with commonly used methods like Newton-Schulz
14 orthogonalization approximation. We theoretically establish an upper bound on
15 these approximation errors, proving their dependence on the condition numbers
16 of moments, conditions we analytically demonstrate are encountered during LLM
17 training. Furthermore, we both theoretically and empirically illustrate that ex-
18 act orthogonalization via SVD substantially improves convergence rates while
19 reducing overall complexity. Empirical evaluations confirm that SUMO acceler-
20 ates convergence, enhances stability, improves performance, and reduces memory
21 requirements by up to 20% compared to state-of-the-art methods.

22 1 Introduction

23 Low-rank gradient-based optimization methods have become powerful tools for reducing memory
24 consumption during the pre-training and fine-tuning of large language models (LLMs), often without
25 sacrificing performance, and sometimes even improving it. For instance, while pre-training LLaMA
26 7B typically requires around 58GB of memory, far exceeding the 24GB available on consumer GPUs
27 like RTX 4090, recent advances, such as those discussed in [1–3], have demonstrated that LLaMA 7B
28 can now be trained from scratch on a single 24GB GPU without the need for costly memory offloading.
29 Theoretical analysis in [1] attributes this efficiency to the inherent low-rank structure of gradients,
30 which allows for optimization within a significantly reduced latent space. Furthermore, [2] found a
31 consistent decrease in gradient rank throughout training, suggesting that low-rank optimization not
32 only reduces memory usage but also converges toward increasingly compact subspaces.

33 However, despite these advancements, existing methods mainly focus on memory savings and often
34 overlook the potential for accelerating convergence. Current approaches typically rely on standard
35 steepest descent techniques and operate under implicit assumptions of isotropic geometry, which
36 can hinder efficiency in ill-conditioned settings. This observation motivates our primary objective:

37 to develop a subspace-aware optimizer that leverages low-rank structure while adapting to the loss
 38 landscape’s geometry. By reevaluating the choice of norm and its influence on gradient descent
 39 dynamics, we aim to design an algorithm that improves generalization, accelerates convergence,
 40 while preserving the memory advantages of low-rank methods.

41 Classical gradient descent, including SGD, performs steepest descent under the Euclidean norm,
 42 which reflects isotropic curvature. However, deep networks exhibit highly anisotropic loss landscapes,
 43 making this assumption suboptimal. Recent work shows that adaptive optimizers like Shampoo [4],
 44 SOAP [5], and Muon [6] can be interpreted as steepest descent under non-Euclidean norms tailored
 45 to network architecture and data structure. As shown in [7], these methods implicitly adapt to spectral
 46 or operator norms, which better capture local curvature and improve convergence. This motivates the
 47 design of subspace-aware optimizers that exploit both low-rank structure and appropriate geometry
 48 to accelerate training.

49 To formalize the role of geometry in optimization, consider a neural network with a differentiable
 50 loss function $\mathcal{L} : \mathcal{W} \rightarrow \mathbb{R}$ defined on a weight space $\mathcal{W} = \mathbb{R}^n$. The local behavior around a point
 51 \mathbf{w} can be approximated by the Taylor expansion, $\mathcal{L}(\mathbf{w} + \Delta\mathbf{w}) \approx \mathcal{L}(\mathbf{w}) + \mathbf{g}^\top \Delta\mathbf{w} + \frac{\lambda}{2} \|\Delta\mathbf{w}\|^2$,
 52 where $\mathbf{g} = \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w})$, $\lambda > 0$ captures the sharpness or curvature of the loss surface and $\|\cdot\|$ is a
 53 chosen norm reflecting the geometry of the optimization landscape. Minimizing this approximation
 54 corresponds precisely to performing steepest descent under the given norm constraint. According to
 55 [8], the solution to this minimization explicitly takes the form,

$$\Delta\mathbf{w} = -\frac{\|\mathbf{g}\|_*}{\lambda} \operatorname{argmax}_{\mathbf{t}: \|\mathbf{t}\|=1} \mathbf{g}^\top \mathbf{t},$$

56 where $\|\cdot\|_*$ denotes the dual norm of $\|\cdot\|$. Adaptive optimizers differ primarily in their norm
 57 choices. Adam utilizes a dynamic Max-of-Max norm constraint. Recent optimizers consider matrix
 58 norms while applying steepest descent at the layer level. Muon imposes a fixed Schatten- p norm
 59 constraint for large p , effectively using the spectral norm on weight matrices [9, 6]. Shampoo [4]
 60 dynamically learns the optimal approximate Schatten- p norm for steepest descent, with its variants
 61 like SOAP [5] applying momentum to efficiently navigate the space of possible norms. Muon, by
 62 contrast, operates within a relatively fixed but large Schatten- p norm, striking a balance between
 63 the dynamic adaptability of Shampoo and the static spectral norm constraints. Since neural network
 64 weights locally act as linear operators on Euclidean spaces, the induced operator (spectral) norm
 65 provides a natural constraint aligning with the curvature characteristics of the loss surface. This
 66 perspective motivates gradient orthogonalization, which ensures optimization updates respect the
 67 spectral norm, inherently controlling the perturbation magnitude and thus enhancing optimization
 68 stability and efficiency [8].

69 While norm-induced optimization methods offer a principled way to align updates with the geometry
 70 of the loss landscape, their practical deployment often incurs substantial computational overhead. For
 71 instance, Shampoo requires computing matrix inverses or root operations at every optimization step,
 72 which can be computationally expensive for large-scale neural networks. Similarly, Muon’s first-
 73 order moments-orthogonalization, though effective, involves an expensive approximation of spectral
 74 decompositions, which is computed by applying five iterations of Newton-Schulz [10] (known also
 75 by (Newton-Schulz5). Therefore, there is an inherent trade-off between the theoretical optimality
 76 provided by these norm-induced optimization approaches and their practical computational demands.

77 To bridge the gap between the geometric advantages of norm-induced methods and their computa-
 78 tional costs, we first analyze the limitations of existing approximations. We derive an upper bound on
 79 the error introduced by the Newton-Schulz orthogonalization, demonstrating that this error increases
 80 with the condition number of the moment matrix. This finding explains the increasing instability
 81 of the Newton-Schulz5 method in ill-conditioned scenarios, which we subsequently demonstrate
 82 to occur in the first-order moment matrices during the training of large language models (LLMs).
 83 Building on this analysis, we establish a convergence rate for Muon optimization and compare it to
 84 an alternative method that replaces the Newton-Schulz approach with exact Singular Value Decom-
 85 position (SVD). Remarkably, we find that the SVD-based approach achieves faster convergence, with
 86 improvements directly proportional to the accumulated errors from the moments orthogonalization
 87 by the Newton-Schulz5 method. Motivated by the empirical observation that gradients in LLMs
 88 often exhibit a low-rank structure, especially during early training, we propose a subspace-aware
 89 optimization scheme. This scheme performs exact SVD-based moment orthogonalization within a
 90 low-dimensional adaptive subspace. This approach benefits from the relatively low computational

cost associated with SVD calculations for low-rank input matrices and enhances the stability of convergence. Also, our approach attains an even greater reduction in memory usage than all previous low-rank training methods by relying solely on first-order moment, as detailed below in Table 1. We support our method with a theoretical convergence guarantee and validate its empirical benefits through experiments, demonstrating faster training and better model performance compared to existing methods.

Table 1: Comparison of properties between SUMO, GaLore, Adam, Shampoo, and SOAP. Assume $\mathbf{W} \in \mathbb{R}^{m \times n}$ with $m \geq n$, a constant projection rank r and a subspace update rate K .

	SUMO	Adam	Shampoo	SOAP	GaLore
Computation	$O(mnr + mn^2/K)$	$O(mn)$	$O(m^3 + n^3)$	$O(m^3 + n^3)$	$O(mnr + mn^2/K)$
Optim. states memory	$nr + mr$	$2mn$	$m^2 + n^2$	$2mn + 2m^2 + 2n^2$	$2nr + mr$
Subspace-aware	✓	×	×	×	✓
Orthogonalization	✓	×	×	×	×

2 Related Work

Low-rank gradient optimization. Low-rank gradients naturally emerge during neural network training, as shown in both theoretical and empirical studies [11–13]. This structure has been leveraged to reduce memory and computational costs during training [14–16]. Recent work [2] showed that gradients in reversible layers [17] tend to collapse to rank one over time and used this to adaptively adjust gradient rank in Adam. In this paper, we show that the same low-rank trend appears in the first-order moment, which we exploit to apply exact orthogonalization efficiently—avoiding the accumulation of approximation errors, such as Newton-Schultz, during optimization.

Memory efficient optimizers. Reducing the memory demands of training large language models (LLMs) has driven extensive algorithmic research. One major approach reduces trainable parameters via low-rank adaptation [18], though such methods often fall short of fully parameterized models, especially during pre-training. Another direction focuses on optimizing training methods, with notable examples including AdaRankGrad, GaLore, Fira, Flora, Adam-mini, GaLore-mini, LDAdam, GoLore, LoQT, and Apollo [2, 1, 19–23, 3], integrating low-rank gradient projections in optimization. In this work, we reduce memory usage even further by relying solely on first-order momentum, as shown in Table 1.

Gradient preconditioning. Preconditioning the Gradient method is critical in enhancing optimizers’ efficiency and effectiveness. Several notable approaches for using a preconditioner have emerged, including methods based on signed gradients [24–27], gradient clipping [28], normalization [28, 29], and gradient whitening [30–32, 6, 33–35]. Recent studies [6, 36] explored gradient orthogonalization strategies, speeding up training. Orthogonalizing gradients effectively constrains updates to lie on directions of uniform magnitude (spectral radius = 1), preventing updates from exaggerating certain gradient directions over others. This procedure ensures a form of normalization that mitigates potential instabilities from ill-conditioned gradients. Unlike these methods, which apply preconditioning or approximate orthogonalization in the high-dimensional parameter space, our approach performs exact SVD-based orthogonalization within an adaptively selected low-rank subspace, offering improved stability and lower computational overhead.

Orthogonal Stochastic Gradient Descent with Momentum (OSGDM). OSGDM [37] is a recently introduced first-order optimization method designed to speed up neural network training by orthogonalizing gradients before the optimization step. Specifically, for a data batch $\xi^{(t)}$ OSGDM applies SVD to the gradient matrix $\mathbf{G}_l = \nabla_{\mathbf{W}_l} \mathcal{L}(\Phi(\xi^{(t)}; \theta))$ of each neural network layer l to generate an orthonormal gradient approximation \mathbf{O}_l . This ensures diversity among learned representations and reduces redundancy. The update rule for OSGDM with momentum term γ and learning rate η is defined as,

$$\mathbf{O}_l^{(t)} = \text{orth}(\mathbf{G}_l), \quad \mathbf{M}_l^{(t+1)} \leftarrow \gamma \mathbf{M}_l^{(t)} + \eta \mathbf{O}_l^{(t)}, \quad \mathbf{W}_l^{(t+1)} \leftarrow \mathbf{W}_l^{(t)} - \mathbf{M}_l^{(t+1)},$$

where $\text{orth}(\mathbf{G}) = (\mathbf{G}\mathbf{G}^\top)^{-1/2} \mathbf{G}$ is the orthogonalization operator, and \mathbf{M}_l is the first order moment of layer l . Despite additional computational overhead from SVD, OSGDM empirically achieves faster convergence and improved accuracy compared to common methods such as Adam.

Muon optimizer. At iteration t , given weight $\mathbf{W}^{(t)}$, momentum μ , learning rate η_t , and objective \mathcal{L}_t , Muon, introduced by [6], constructs the update rule,

$$\mathbf{M}^{(t)} = \mu \mathbf{M}^{(t-1)} + \mathbf{G}_l^{(t)}, \quad \mathbf{O}_l^{(t)} = \text{Newton-Schulz5}(\mathbf{M}^{(t)}), \quad \mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta_t \mathbf{O}_l^{(t)}.$$

Here, $\mathbf{M}^{(t)}$ is the momentum at iteration t , initialized as a zero matrix when $t = 0$. The Newton-Schulz5 method [33] approximates $(\mathbf{M}^{(t)} \mathbf{M}^{(t)\top})^{-1/2} \mathbf{M}^{(t)}$, orthogonalizing $\mathbf{M}^{(t)}$ and thus ensuring uniform update directions, avoiding dominance by few directions. Muon explicitly controls the norm of gradient updates—particularly the spectral norm (or Schatten- p norm), which limits updates to smaller, well-conditioned steps in parameter space. By constraining the spectral norm, moment orthogonalization implicitly prevents overly large or ill-conditioned parameter updates. Such updates often lead to poor generalization due to instability or overfitting. Shortly after the introduction of Muon, the study in [38] proposed a framework to scale Muon for larger LLMs, mainly adding weight decay, and carefully adjusting the per-parameter update scale.

3 Method and Main Results

3.1 Theoretical Motivation: Exact moments orthogonalization leads to significantly faster convergence

Previous work on pre-training and fine-tuning large language models (LLMs) has primarily focused on reducing memory usage for constrained hardware [] or lowering computational cost []. In this paper, we take a step toward accelerating LLM optimization by showing that applying exact orthogonalization (e.g., via SVD) to the first-order moment offers a practical advantage, even over the most accurate approximations, such as the commonly used Newton-Schulz5 method. Specifically, we find that SVD yields faster convergence and lower computational overhead. To support this, we first present a new observation: the moment matrix in LLM training tends to decrease in rank over time. Building on this, we then derive an upper bound on the approximation error of Newton-Schulz5, showing that it depends on both the number of iterations and the matrix condition number, highlighting its limitations in ill-conditioned or low-rank settings (which is exactly the case in LLM optimization moments). This motivates the need for more accurate orthogonalization of moment matrices during LLM training. Of course, applying SVD directly to full-sized layers is generally impractical. The surprising result, however, is that when integrated into a low-rank optimization scheme, the use of SVD becomes not only feasible but preferable. We conclude with a convergence analysis of Muon optimization, which, under these conditions, converges significantly more slowly than the SVD-based alternative. To the best of our knowledge, our convergence analysis of Muon optimization is the first to avoid neglecting the error in the Newton-Schulz approximation [39]. The proofs of all lemmas and theorems of this section are relegated to the Appendix A.

Lemma 3.1 (Moment Becomes Low-Rank During Training). *Let $\mathbf{M}^{(t)} \in \mathbb{R}^{n \times m}$ denote the first moment of a reversible layer¹ in a moment-based optimization algorithm, updated according to $\mathbf{M}^{(t)} = \beta_1 \mathbf{M}^{(t-1)} + \mathbf{G}^{(t)}$, where $\mathbf{G}^{(t)}$ is the gradient matrix at iteration t . Let $\mathbf{M}^{(t)} = \mathbf{U}^{(t)} \Sigma^{(t)} \mathbf{V}^{(t)\top}$ be the singular value decomposition (SVD) of $\mathbf{M}^{(t)}$, and define the rank- r orthogonal projection matrix as $\mathbf{P}^{(t)}(r) = \mathbf{U}^{(t)}[:, 1:r] \mathbf{U}^{(t)}[:, 1:r]^\top$. Then the relative error of the best rank-one approximation,*

$$\kappa_M(t) \triangleq \frac{\|\mathbf{M}^{(t)} - \mathbf{P}^{(t)}(1) \mathbf{M}^{(t)}\|_F^2}{\|\mathbf{M}^{(t)}\|_F^2}, \quad (1)$$

satisfies $\kappa_M(t) \leq O(C^{-t})$ for some constant $C > 1$.

The above result, in (1), implies that $\mathbf{M}^{(t)}$ approaches its rank-one approximation $\mathbf{P}^{(t)}(1) \mathbf{M}^{(t)}$, as the iteration number increases, namely, $\mathbf{M}^{(t)}$ becomes rank-one. The following Lemma 4 characterizes the impact of the moments’ low-rank structure on the approximation error of the Newton-Schulz5 orthogonalization.

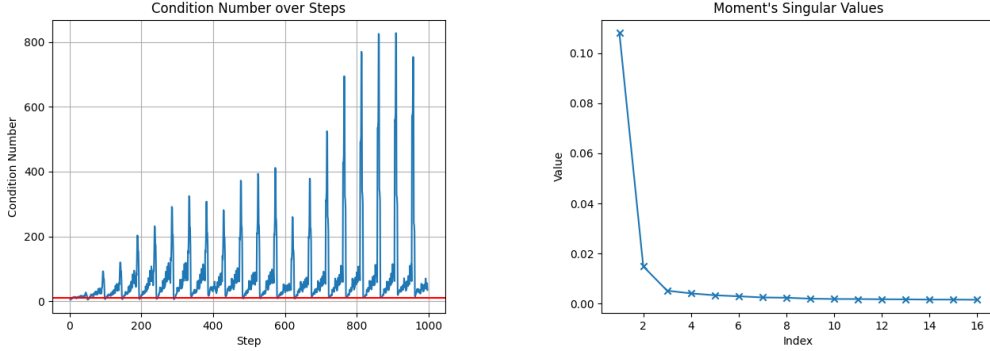
Lemma 3.2 (Orthogonalization error \mathcal{E}_i). *For a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, let σ_1 be the largest singular value of $\mathbf{A} \mathbf{A}^\top$ and σ_m be the smallest (without the loss of generality, assume $m \leq n$). Let $r \leq m$ be*

¹Reversible networks are formally defined in Appendix B.1

169 the largest index where $\sigma_r > \sigma_{r+1} = \dots \sigma_m \geq 0$. Let $\kappa = \frac{\sigma_1}{\sigma_m}$ by the condition number of $\mathbf{A}\mathbf{A}^\top$.
 170 Denote \mathcal{E}_i the error of Newton-Schultz after i iterations. Then we have

$$\|\mathcal{E}_i\|_F \leq \sqrt{r} \cdot \left(1 - \frac{1}{\kappa}\right)^{2^i}. \quad (2)$$

171 According to the lemma, the approximation error grows exponentially with the condition number.
 172 Given the low-rank structure of the first-order moments, low-dimensional optimization offers a
 173 means to mitigate this error. Specifically, projecting the moment estimates $\hat{\mathbf{M}}^{(t)}$ onto their dominant
 174 (small) r -dimensional subspace ensures that the squared moment $\hat{\mathbf{M}}^{(t)}\hat{\mathbf{M}}^{(t)\top}$ is constructed using
 175 only the top r squared eigenvalues. These dominant components are significantly larger and exclude
 176 near-zero values, resulting in a substantially lower condition number compared to that of the full-rank
 177 squared moment matrix. This observation motivates the use of the Muon optimizer within a low-rank
 178 optimization framework for LLMs, including 2D reversible layers. Such an approach not only
 179 preserves the inherent memory efficiency of low-rank methods but also reduces the approximation
 180 error in the optimization step, potentially leading to faster convergence and improved performance
 181 over full-dimensional training. However, we also empirically observe that the eigenvalues of the
 182 moment matrix decay gradually. As shown in Figure 1, even when projecting onto the dominant
 183 subspace, the resulting matrix $\hat{\mathbf{M}}^{(t)}\hat{\mathbf{M}}^{(t)\top}$, composed of the top $r = 16$ squared eigenvalues, can
 184 still exhibit a large condition number, thereby introducing non-negligible approximation error.



(a) Condition number of the first-order moment vs. training step. The red line marks value 10.

(b) Illustration of the moment's singular value decay, taken arbitrarily at step 100.

Figure 1: Evidence of anisotropy and ill-conditioning in the first-order moment matrix as a function of the Galore steps of the Roberta-base model [40] on the GLUE dataset RTE task [41]: (a) condition number growth, (b) spectral decay of moment.

185 To comprehend the cumulative error of Newton-Schulz5 orthogonalization at each optimization step,
 186 we proceed to derive the convergence rate of the Moun optimization. To that end, we now provide
 187 some notations. Consider a neural network denoted as $\Phi(\cdot; \theta)$, which consists of L layers and is
 188 parameterized by $\theta \triangleq [\mathbf{W}_1^{d_1 \times d_0}, \dots, \mathbf{W}_{L-1}^{d_{L-1} \times d_{L-2}}, \mathbf{W}_L^{d_L \times d_{L-1}^{L-1}}]$. Here, \mathbf{W}_i represents the weights
 189 tensor parameters associated with the i -th layer, for $i \in [L]$. We denote the differential loss \mathcal{L} , where
 190 with a slight abuse of notation, we write the training problem by $\min_{\mathbf{W}} \mathcal{L}(\mathbf{W}) = \mathbb{E}_{\xi}[\mathcal{L}(\Phi(\mathbf{W}, \xi))]$,
 191 if the context refers to the weights of a certain layer. We use the Frobenius norm, denoted $\|\cdot\|_F$,
 192 which is induced by the inner product $\langle \mathbf{X}, \mathbf{Y} \rangle = \text{tr}(\mathbf{X}^\top \mathbf{Y})$. Assume that the stochastic gradient
 193 $\nabla \mathcal{L}(\mathbf{W}, \xi)$ is an unbiased estimator of the full gradient $\nabla \mathcal{L}(\mathbf{W})$, with variance bounded by σ^2 , i.e.,
 194 $\mathbb{E}[\|\nabla \mathcal{L}(\mathbf{W}, \xi) - \nabla \mathcal{L}(\mathbf{W})\|_F^2] \leq \sigma^2$. Let $\mathcal{E}_i^{(t)} = \text{orth}(\mathbf{M}^{(t)}) - \text{Newton-Schulz}(\mathbf{M}^{(t)})$ denote the
 195 approximation error of the Newton-Schulz (with $i \geq 1$ iteration) at time t , where $\mathbf{M}^{(t)}$ denotes the
 196 moment at iteration t .

197 **Lemma 3.3** (Exact convergence rate of Muon). *Consider the Muon optimizer update defined by*

$$\begin{aligned} \mathbf{M}^{(t)} &\leftarrow \beta \mathbf{M}^{(t-1)} + (1 - \beta) \mathbf{G}^{(t)}, \\ \mathbf{O}^{(t)} &\leftarrow \mathbf{U}^{(t)} \mathbf{V}^{(t)\top} + \mathcal{E}_i^{(t)}, \quad (i \text{ iterations Newton-Schulz approximation}), \end{aligned}$$

$$\mathbf{W}^{(t+1)} \leftarrow \mathbf{W}^{(t)} - \eta_t \mathbf{O}^{(t)},$$

198 where $\mathbf{M}^{(t)} = \mathbf{U}^{(t)} \mathbf{S}^{(t)} \mathbf{V}^{(t)\top}$ denotes the singular value decomposition of $\mathbf{M}^{(t)}$, and $\mathcal{E}_i^{(t)}$ represents
199 the Newton-Schulz5 approximation error. Suppose the following:

- 200 • The gradient $\nabla \mathcal{L}(\mathbf{W})$ is L -Lipschitz continuous.
- 201 • There exists $\delta > 0$ such that $\|\mathcal{E}_i^{(t)}\| \leq \delta \|\mathbf{U}_t \mathbf{V}_t^\top\| = \delta \sqrt{n}$, for all t .

If we take $\beta = 1 - \alpha$ with $\alpha = \min(\frac{\sqrt{RL}}{\sigma\sqrt{T}}, 1)$, $\eta_t = \eta = \frac{\sqrt{4R}}{(10/(1-\beta) + 2n + 4n\delta + 2n\delta^2)TL}$, and $B = 1$
(batch free convergence) then $\frac{1}{T} \sum_{t=1}^T \mathbb{E} [\|\nabla \mathcal{L}(\mathbf{W}^{(t)})\|]$ is bounded by

$$\mathcal{O} \left(\left[\frac{\sqrt{RLn(2 + 4\delta + 2\delta^2)}}{\sqrt{T}} + \frac{\sigma^2}{\sqrt{RLT}} + \frac{\sigma(RL)^{1/4} + \sqrt{\sigma}(RL)^{1/4}}{T^{1/4}} \right] \frac{1}{1 - 4\sqrt{n}\delta} \right),$$

202 where $R = \mathcal{L}(\mathbf{W}^{(0)}) - \mathcal{L}^*$. If we take β as an arbitrary constant, we have to take $B = T$, and we
203 have,

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \|\nabla \mathcal{L}(\mathbf{W}^{(t)})\| \leq \mathcal{O} \left(\left[\frac{\sqrt{RLn(2 + 4\delta + 2\delta^2)}}{\sqrt{T}} + \frac{\sqrt{RL}}{\sqrt{T}} + \frac{\sigma}{T^{3/2}} + \frac{\sigma}{\sqrt{T}} \right] \frac{1}{1 - 4\sqrt{n}\delta} \right).$$

Remark 3.4 (Comparison: slower convergence vs exact orthogonalization). When $\delta = 0$, indicating
an absence of error, the convergence rate is aligned with the one derived in [42], Theorem 2.1, that is

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} [\|\nabla \mathcal{L}(\mathbf{W}^{(t)})\|] \leq \mathcal{O} \left(\frac{\sqrt{nRL}}{\sqrt{T}} + \frac{\sigma}{T^{3/2}} + \frac{\sigma}{\sqrt{T}} \right).$$

204 This result overlooks the error associated with the Newton-Schulz5 approximation because it is
205 based on a theoretically exact method of orthogonalization.

206 *Remark 3.5* (The impact of δ on the convergence rate). A reduction in δ is associated with an
207 improvement in the convergence rate. Furthermore, it should be noted that δ influences the step size
208 η ; a larger δ results in a smaller step size, providing an additional explanation for the convergence
209 rate.

210 *Remark 3.6* (The size of δ). We acknowledge that the findings of our analysis are applicable only
211 under the conditions specified in $1 - 4\sqrt{n}\delta > 0 \Rightarrow \delta < \frac{1}{4\sqrt{n}}$. In scenarios where $\delta > \frac{1}{4\sqrt{n}}$ applies,
212 the algorithm may fail to converge. To ensure that δ remains sufficiently small, the Newton-Schulz5
213 method necessitates a substantial number of iterations, consequently slowing down the convergence.

214 *Remark 3.7* (Speed-up by SVD vs Newton-Schulz5 approximation). According to Lemma 3.2, these
215 low-rank moments, which inherently possess exceptionally high κ , result in an error expressed by
216 $(1 - \varepsilon)^{2^i}$ concerning a remarkably small ε . This situation necessitates numerous iterations for the
217 Newton-Schulz method to converge. For example, if $(1 - \varepsilon) = 0.99$ is considered and Newton-
218 Schulz5 is utilized with 5 iterations, the error would be $\approx 0.99^{32} = 0.725$, relative to the norm of
219 the moment, namely \mathbf{M} .

220 Recall that in the low-rank setting, accurately computing the pseudoinverse using singular value
221 decomposition (SVD) is numerically advantageous and reasonably computationally affordable compared
222 to iterative methods such as Newton-Schulz. For a general matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$, the SVD
223 provides a decomposition $\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top$, with $\mathbf{U} \in \mathbb{R}^{n \times n}$, $\mathbf{\Sigma} \in \mathbb{R}^{n \times m}$, and $\mathbf{V} \in \mathbb{R}^{m \times m}$. The Moore-
224 Penrose pseudoinverse is then calculated as $\mathbf{A}^\dagger = \mathbf{V} \mathbf{\Sigma}^\dagger \mathbf{U}^\top$, requiring approximately $4nm^2 + 8m^3$
225 floating-point operations (FLOPs) for the initial decomposition, and an additional $mn^2 + m^2n$ FLOPs
226 for subsequent multiplications, totaling roughly $4nm^2 + 8m^3 + mn^2 + m^2n$ FLOPs.

227 Alternatively, approximating the inverse of $\mathbf{A}^\top \mathbf{A} \in \mathbb{R}^{m \times m}$ using Newton-Schulz iterations involves
228 nm^2 FLOPs to form the matrix $\mathbf{A}^\top \mathbf{A}$, approximately $20m^3 + 10m^2$ FLOPs for five iterations, and
229 an additional m^2n FLOPs to multiply by \mathbf{A}^\top , resulting in a total of about $nm^2 + m^2n + 20m^3 +$
230 $10m^2$ FLOPs. For example, when the rank is $m = 8$ and $n = 1024$, the SVD approach requires
231 approximately twice the number of operations compared to Newton-Schulz5. Nonetheless, given the
232 superior numerical stability and inherent optimality of the SVD-based method, this moderate increase
233 in computational effort remains acceptable, especially when accuracy and stability are prioritized.

234 3.2 Method

235 We are now ready to present our main algorithm designed to accelerate the low-rank optimization
 236 scheme outlined in Algorithm 1. A detailed mathematical formulation of the weight update rule
 237 proposed in this paper can be found in Appendix C. The algorithm consists of four primary blocks, all
 238 contained within an outer loop that continues until convergence is achieved or a predefined number
 of epochs is reached. Each block serves a specific purpose, which will be explained in detail below.

Algorithm 1 SUMO: Subspace-Aware Moment-Orthogonalization Optimization

Input: A weight matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$ with $m \geq n$. Step size η , scale factor α , decay rates $\{\beta_1, \beta_2\}$,
 weight decay λ , rank r , subspace update frequency K , small number $k \in \mathbb{N}$, step clipping ratio γ .
Initialize: $t \leftarrow 0$
repeat
 # Block 1: Calculate low rank gradient projection.
 Sample mini-batch $B = \{\xi_1, \xi_2, \dots, \xi_{|B|}\}$
 Compute $\mathbf{G}^{(t)} \leftarrow \sum_{i=1}^{|B|} \frac{\partial}{\partial \mathbf{W}} \mathcal{L}(\Phi(x_i, \boldsymbol{\theta}), y_i)$
 if $t \bmod K = 0$ **then**
 $\mathbf{Q}_t \leftarrow \text{Truncated_Randomized_SVD}(\mathbf{G}_t)$ # Alternatively $\text{Truncated_SVD}(\mathbf{G}_t)$
 # Block 1.1: Moment subspaces transformation
 $\mathbf{R}^{r \times r} \leftarrow \mathbf{Q}^{(t)\top} \mathbf{Q}^{(t-1)}$ if $t \geq 1$, else $\mathbf{0}^{r \times r}$
 $\mathbf{M}^{(t)r \times n} \leftarrow \mathbf{R} \mathbf{M}^{(t-1)}$, if $t \geq 1$, else $\mathbf{0}^{r \times n}$ {1st-order moment}
 end if # Alternatively criteria $\|\hat{\mathbf{G}}_t\| \leq \varsigma$
 $\hat{\mathbf{G}}^{(t)} \leftarrow \mathbf{Q}^{(t)\top} \mathbf{G}^{(t)}$
 # Block 2: Low-rank steepest-decent step (moment orthogonalization)
 $\mathbf{M}^{(t)} \leftarrow \mu \mathbf{M}^{(t-1)} + \hat{\mathbf{G}}^{(t)}$
 $\mathbf{O}^{(t)} \leftarrow \text{Orthogonalization_SVD}(\mathbf{M}^{(t)})$
 # Block 3 (Optional):
 if $\frac{\|\mathbf{O}^{(t)}\|}{\|\mathbf{O}^{(t-1)}\|} > \gamma$ **then** $\mathbf{O}^{(t)} \leftarrow \frac{\mathbf{O}^{(t)}}{\|\mathbf{O}^{(t)}\|} \cdot \gamma \|\mathbf{O}^{(t-1)}\|$
 # Block 4: Update weight in original space.
 $\mathbf{W}^{(t)} \leftarrow \mathbf{W}^{(t-1)} - \alpha \eta \mathbf{Q}^{(t)} \mathbf{O}^{(t)} - \eta \cdot \lambda \mathbf{W}^{(t-1)}$
 $t \leftarrow t + 1$
until convergence criteria met (e.g. epoch number, gradient norm $\|\mathbf{G}_t\| \leq \xi$)
return $\mathbf{W}^{(T)}$

- 239
- 240 • **Block 1:** We select the subspace along the directions of the r largest eigenvectors, but since
 241 computing full SVD for large matrices is computationally intensive and memory-demanding,
 242 we leverage the Randomized-SVD by [43], which is an efficient technique for producing a
 243 “good” proxy for the optimal low-rank approximation. It solves the optimization problem
 244 $\arg \min_{\mathbf{Q} \in \mathbb{R}^{n \times r}} \|\mathbf{G} - \mathbf{Q} \mathbf{Q}^\top \mathbf{G}\|_F$, and approximates the matrix \mathbf{G} as $\mathbf{G}_{\text{app},r} \approx \mathbf{Q} \mathbf{Q}^\top \mathbf{G}$, that
 245 requires $O(mnr + mr^2)$ operations, instead of $O(\min(mn^2, m^2n))$ applied by SVD.
 - 246 • **Block 1.1:** We transform the first-order moments evaluated during the low-rank optimization
 247 steps, which occur in Block 2, between the preceding and the newly updated subspace. This
 248 transformation is required because, as will be demonstrated later, within Block 2, the first moments
 249 of the gradients are aligned with the previously projected subspace. Consequently, a transformation
 250 is necessary to translate them from the former subspace to the current one.
 - 251 • **Block 2:** Here we calculate the (steepest) optimization step. SVD operation is adopted to solve
 252 exactly $(\mathbf{M}^{(t)} \mathbf{M}^{(t)\top})^{-1/2} \mathbf{M}^{(t)}$. Let $\mathbf{U} \Sigma \mathbf{V}^\top = \hat{\mathbf{M}}^{(t)}$ be the singular value decomposition (SVD)
 253 of $\hat{\mathbf{M}}^{(t)}$, we will have $(\mathbf{M}^{(t)} \mathbf{M}^{(t)\top})^{-1/2} \mathbf{M}^{(t)} = \mathbf{U} \mathbf{V}^\top$, which orthogonalizes $\hat{\mathbf{M}}^{(t)}$.
 - 254 • **Block 3:** Rather than using standard gradient clipping, we adopt the Norm-growth Limiter (NL)
 255 introduced by [19], which has been shown to slightly outperform traditional clipping techniques
 256 by better constraining the progression of gradient magnitudes. Specifically, the gradient update is
 257 modified as follows, if $\frac{\|\mathbf{O}^{(t)}\|}{\|\mathbf{O}^{(t-1)}\|} > \gamma$ then $\mathbf{O}^{(t)} \leftarrow \frac{\mathbf{O}^{(t)}}{\|\mathbf{O}^{(t)}\|} \cdot \gamma \|\mathbf{O}^{(t-1)}\|$, where the scalar γ serves

258 as a growth threshold to regulate abrupt increases in gradient norm from one iteration to the next.
 259 We use $\gamma = 1.1$, which empirically leads to the highest results.

260 • **Block 4:** The pre-trained model parameters are updated using the low-dimensional back-projection
 261 matrix along with weight decay. To ensure stable training across parameter matrices of different
 262 shapes, we interpret the root mean square (RMS) magnitude of updates as implicit *layer-wise*
 263 *learning rate adaptation*, following the approach in [38]. By scaling updates with $\sqrt{\max(m, n)}$,
 264 our method compensates for shape-induced differences in magnitude, achieving consistent effective
 265 learning rates across layers, similar to adaptive optimizers like AdamW.

266 For clarity, we can assume, without loss of generality, that $m \geq n$. In the opposite scenario, the
 267 projection matrix would multiply the gradient from the right-hand side.

268 **Theorem 3.8** (Convergence of SUMO). *For a loss function \mathcal{L} , and given architecture Φ , suppose that*
 269 *the compositions of $f \equiv \mathcal{L}(\Phi(\cdot))$ is β -smooth non-convex function that is bounded by some $M \in \mathbb{R}_+$.*
 270 *Let $\mathbf{G}_j^{(t)}$ denote the gradient matrix w.r.t. the j -th reversible layer \mathbf{W}_t^j , at time $t \in \mathbb{N}$, for all $j \in [L]$*
 271 *and $t \in \mathbb{N}$, and $T_\ell, \ell \in \mathbb{N}$ times are set by a convergence criterion (that is, $\|\hat{\mathbf{G}}_{T_\ell}\| \leq \varsigma_\ell$). Then, there*
 272 *exist $C \in \mathbb{R}_+$ and N such that for all $T_N > \frac{C}{\varepsilon^2}$, and $\frac{1}{T_N} \sum_{i=0}^{N-1} \sum_{t=T_i}^{T_{i+1}-1} \|\mathbf{G}_j^{(t)}\|_F^2 \leq \varepsilon$. Namely,*
 273 *Algorithm 1 achieves an ε -critical point,² i.e., $\|\mathbf{G}_j^{(t)}\|_F^2 \leq \varepsilon$, for some $t \in \mathbb{N}$, and any $j \in [L]$.*

274 The proof of Theorem 1 can be found in Appendix A. We emphasize that the convergence proof
 275 for Galore in [1] addresses only optimization within fixed subspaces, ignoring dynamic updates.
 276 AdaRankGrad’s proof [2] first established guarantees for the complete dynamic-subspace updates,
 277 yet both prior works simplified the inner steps as standard SGD. In contrast, SUMO’s convergence
 278 proof explicitly considers the exact optimization steps without simplifications.

279 To reduce memory consumption, Algorithm 1 applies per-layer weight updates during backpropaga-
 280 tion, following recent works such as [44]. This contrasts with conventional optimizers that store full
 281 gradients and update all weights afterward, leading to potential inefficiencies. Details for post-hoc
 282 adapter extraction are discussed in Appendix B.

283 4 Experiments

284 **Fine-tuning on GLUE benchmark.** Our
 285 model was evaluated using the GLUE bench-
 286 mark [41] through the fine-tuning of the pre-
 287 trained Roberta-base model [40] across eight
 288 tasks. The comparative analysis includes full
 289 fine-tuning, LoRA, and GaLore methodologies,
 290 with the results enumerated in Table 2. The
 291 metrics reported are the overall (matched and
 292 mismatched) accuracy for MNLI, Matthew’s
 293 correlation for CoLA, Pearson correlation for
 294 STS-B, F1-score for MRPC, and accuracy for
 295 the remaining tasks. Evidently, our approach
 296 enhances the fine-tuning accuracy while requir-
 297 ing less training memory, utilizing only a single
 298 moment compared to GaLore. The experiments
 299 were carried out using the NVIDIA A100 GPU.

300 **Pre-training LLAMA on C4 Dataset.** To
 301 demonstrate the benefits of our method for pre-
 302 training LLAMA by following the evaluation
 303 form [1]. Specifically, we compare the performance of SUMO to state-of-the-art methods in terms
 304 of perplexity and memory efficiency. For this evaluation, we trained large LLaMA-based models
 305 on the C4 dataset, a curated and extensive version of the Common Crawl web corpus [45]. This

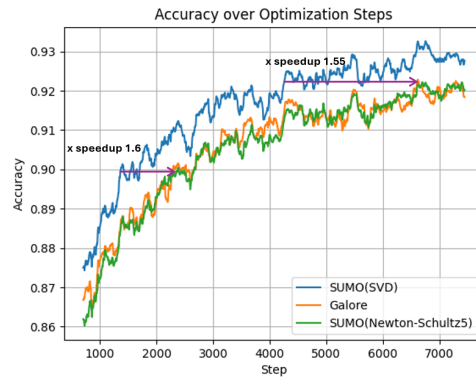


Figure 2: SUMO with SVD demonstrates superior convergence speed (~ 1.6 faster), attaining comparable or higher accuracy than GaLore and SUMO with Newton-Schultz5 with significantly fewer optimization steps on QNLI.

²Also known as ε -stationary, see, e.g., [12].

Table 2: Evaluation comparison of SUMO against state-of-the-art memory-efficient fine-tuning methods on the GLUE benchmark using the pre-trained RoBERTa-Base model. For comparison, we provide detailed results for SUMO using both SVD and Newton-Schulz5 orthogonalizations (ablation study).

Model	Memory	CoLA	STS-B	MRPC	RTE	SST2	MNLI	QNLI	QQP
Full Fine-Tuning	747M	62.24	90.92	91.30	79.42	94.57	87.18	92.33	92.28
LoRA (rank=4)	257M	61.38	90.57	91.07	78.70	92.89	86.82	92.18	91.29
GaLore (rank=4)	253M	60.35	90.73	92.25	79.42	94.0	87.0	92.24	91.06
SUMO (Newton-Schulz5, rank=4)	197M	61.8	90.82	92.43	79.36	94.17	86.92	92.26	91.27
SUMO (SVD, rank=4)	197M	62.3	91.04	93.5	81.07	94.93	87.34	93.26	91.68
LoRA (rank=8)	264M	61.83	90.80	91.90	79.06	93.46	86.94	92.25	91.22
GaLore (rank=8)	257M	60.06	90.82	92.0	79.78	94.38	87.17	92.2	91.11
SUMO (Newton-Schulz5, rank=4)	198M	61.74	90.79	91.94	79.69	94.17	87.21	92.24	91.38
SUMO (rank=8)	198M	61.7	91.1	93.7	81.37	94.82	87.58	93.67	91.72

dataset is widely used for pre-training language models and developing word representations. To better reflect real-world pre-training scenarios, we conducted training on a non-repeating, large-scale dataset and scaled model sizes up to 350 million parameters. The results of these experiments are shown in Table 3. Experiments were conducted using an NVIDIA H200 GPU.

Table 3: Comparison of state-of-the-art low-rank algorithms for pre-training LLaMA models of varying sizes on the C4 dataset. The results are reported in terms of validation perplexity. As can be seen, SUMO leads to improved performance with a substantial memory reduction compared to leading parameter-efficient fine-tuning schemes.

Method	60M	130M	350M	1B
Full-Rank	34.06 (0.36G)	25.08 (0.76G)	18.80 (2.06G)	15.56(7.80G)
GaLore	34.88 (0.24G)	25.36 (0.52G)	18.95 (1.22G)	15.64(4.38G)
Low-Rank	78.18 (0.26G)	45.51 (0.54G)	37.41 (1.08G)	142.53(3.57G)
LoRA	34.99 (0.36G)	33.92 (0.80G)	25.58 (1.76G)	19.21(6.17G)
ReLoRA	37.04 (0.36G)	29.37 (0.80G)	29.08 (1.76G)	18.33(6.17G)
SUMO	34.26 (0.23G)	24.87 (0.51G)	18.69 (1.16G)	14.68 (1.16G)
Training Tokens	1.1B	2.2B	6.4B	13.1B
r/d_{model}	128/256	256/768	256/1024	512/2048

Few/Zero-shot reasoning and long-context generalization. To evaluate our method’s performance on a complex reasoning task, we use the GSM8K dataset [46], testing systematic generalization. For these experiments, we used a batch size of 32 and 10 epochs for fine-tuning. We present the performance result in Table 4 training Phi-2 (2.7B) model [47], and in Table 5 training Lamma (1B) model [48]. The results demonstrate that the proposed method significantly improves generalization to out-of-distribution data. The experiments were conducted on an NVIDIA H200 GPU.

Table 4: Zero-shot evaluation on GSM8K dataset (Phi-2, 2.7B).

Phi-2 (2.7B)	Rank	Accuracy (0-shot)
Base Model	64	15.16%
Galore	64	52.24%
LoRA	64	42.8%
SUMO	64	54.13%

Table 5: 8-shot evaluation on GSM8K dataset (LLaMA, 3B).

LLaMA (3B)	Rank	Accuracy (8-shot)
Base Model	64	17.93%
Galore	64	74.9%
LoRA	64	68.3%
SUMO	64	76.7%

Additional experiments and ablation studies are presented in the Appendix D. For all experiments, we used $\lambda = 0.5$, $\mu = 0.95$ and $\gamma = 1.1$.

5 Discussion

Our results highlight that exact moment orthogonalization within a low-dimensional adaptive subspace significantly improves both convergence and stability in memory-efficient LLM training. By avoiding the approximation errors of Newton-Schulz5, SUMO leverages the low-rank structure of gradients to enable accurate, spectral-norm-aligned updates with minimal overhead.

Empirically, SUMO outperforms prior low-rank methods in both fine-tuning and pretraining tasks, achieving better performance in memory consumption reduction compared to the memory-efficient benchmarks such as Galore. Our theoretical analysis further confirms its superior convergence properties under practical conditions. These findings position SUMO as a simple yet effective alternative to approximate geometric optimizers.

References

- [1] Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection, 2024.
- [2] Yehonathan Refael, Jonathan Svirsky, Boris Shustin, Wasim Huleihel, and Ofir Lindenbaum. Adarankgrad: Adaptive gradient rank and moments for memory-efficient LLMs training and fine-tuning. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [3] 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. *arXiv preprint arXiv:2412.05270*, 2024.
- [4] Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization, 2018.
- [5] Hao Sun, Li Shen, Qihuang Zhong, Liang Ding, Shixiang Chen, Jingwei Sun, Jing Li, Guangzhong Sun, and Dacheng Tao. Adam: Boosting sharpness-aware minimization with adaptive learning rate and momentum for training deep neural networks, 2023.
- [6] Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cecista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024.
- [7] Louis Cesista Franz. The Case for Muon, October 2024.
- [8] Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. *arXiv preprint arXiv:2409.20325*, 2024.
- [9] Dmitry Kovalev. Understanding gradient orthogonalization for deep learning via non-euclidean trust-region optimization, 2025.
- [10] Nicholas J Higham. Newton’s method for the matrix square root. *Mathematics of computation*, 46(174):537–549, 1986.
- [11] Jingzhao Zhao, Frederik T. Schaefer, and Anima Anandkumar. Zero initialization: Initializing neural networks with only zeros and ones. *Transactions on Machine Learning Research*, 2022.
- [12] Romain Cosson, Ali Jadbabaie, Anuran Makur, Armin Reiszadeh, and Devavrat Shah. Low-Rank Gradient Descent. *IEEE Open Journal of Control Systems*, 2023.
- [13] Greg Yang, Jacob B. Simon, and Jeremy Bernstein. A spectral condition for feature learning. *arXiv preprint arXiv:2310.17813*, 2023. *arXiv:2310.17813*.
- [14] M. Gooneratne, K. C. Sim, P. Zadrazil, A. Kabel, F. Beaufays, and G. Motta. Low-rank gradient approximation for memory-efficient on-device training of deep neural network. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. IEEE, 2020.
- [15] Shuang Huang, Brian D. Hoskins, Michael W. Daniels, Matthew D. Stiles, and George C. Adam. Low-Rank Gradient Descent for Memory-Efficient Training of Deep In-Memory Arrays. *ACM Journal on Emerging Technologies in Computing Systems*, 2023.
- [16] Ionut-Vlad Modoranu, Alexander Kalinov, Ermin Kurtic, Erwin Frantar, and Dan Alistarh. Error Feedback Can Accurately Compress Preconditioners. *ArXiv preprint arXiv:2306.06098*, 2023. *arXiv:2306.06098*.
- [17] Yuandong Tian, Lantao Yu, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning with dual deep networks, 2021.
- [18] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.
- [19] 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? *arXiv preprint arXiv:2410.01623*, 2024.

- [20] Yongchang Hao, Yanshuai Cao, and Lili Mou. Flora: Low-rank adapters are secretly gradient compressors. *ArXiv*, abs/2402.03293, 2024.
- [21] Yushun Zhang, Congliang Chen, Ziniu Li, Tian Ding, Chenwei Wu, Yinyu Ye, Zhi-Quan Luo, and Ruoyu Sun. Adam-mini: Use fewer learning rates to gain more. *arXiv preprint arXiv:2406.16793*, 2024.
- [22] Thomas Robert, Mher Safaryan, Ionut-Vlad Modoranu, and Dan Alistarh. Ldadam: Adaptive optimization from low-dimensional gradient statistics, 2025.
- [23] Sebastian Loeschke, Mads Tofttrup, Michael J. Kastoryano, Serge Belongie, and Vésteinn Snæbjarnarson. Loqt: Low-rank adapters for quantized pretraining, 2024.
- [24] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR, 2018.
- [25] Michael Crawshaw, Mingrui Liu, Francesco Orabona, Wei Zhang, and Zhenxun Zhuang. Robustness to unbounded smoothness of generalized signsgd. *Advances in neural information processing systems*, 35:9955–9968, 2022.
- [26] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery of optimization algorithms. In *NeurIPS*, 2023.
- [27] Frederik Kunstner, Jacques Chen, Jonathan Wilder Lavington, and Mark Schmidt. Noise is not the main factor behind the gap between sgd and adam on transformers, but sign descent might be. *arXiv preprint arXiv:2304.13960*, 2023.
- [28] Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 33:15383–15393, 2020.
- [29] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.
- [30] Zhirong Yang and Jorma Laaksonen. Principal whitened gradient for information geometry. *Neural Networks*, 21(2-3):232–240, 2008.
- [31] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.
- [32] Dongseong Hwang. Fadam: Adam is a natural gradient optimizer using diagonal empirical fisher information. *arXiv preprint arXiv:2405.12807*, 2024.
- [33] Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology, 2024.
- [34] Jeremy Bernstein and Laker Newhouse. Modular duality in deep learning, 2024.
- [35] David E Carlson, Edo Collins, Ya-Ping Hsieh, Lawrence Carin, and Volkan Cevher. Preconditioned spectral descent for deep learning. *Advances in neural information processing systems*, 28, 2015.
- [36] Mark Tuddenham, Adam Prügel-Bennett, and Jonathan Hare. Orthogonalising gradients to speed up neural network optimisation. *arXiv preprint arXiv:2202.07052*, 2022.
- [37] Mark Tuddenham, Adam Prügel-Bennett, and Jonathan Hare. Orthogonalising gradients to speed up neural network optimisation, 2022.
- [38] Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is scalable for llm training, 2025.

- 421 [39] Jiaxiang Li and Mingyi Hong. A note on the convergence of muon and further. *arXiv preprint*
422 *arXiv:2502.02900*, 2025.
- 423 [40] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike
424 Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining
425 approach. *arXiv preprint arXiv:1907.11692*, 2019.
- 426 [41] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix
427 Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose
428 language understanding systems. In *Advances in Neural Information Processing Systems*,
429 volume 32, 2019.
- 430 [42] Jiaxiang Li and Mingyi Hong. A note on the convergence of muon and further, 2025.
- 431 [43] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness:
432 Probabilistic algorithms for constructing approximate matrix decompositions, 2010.
- 433 [44] Kai Lv, Hang Yan, Qipeng Guo, Haijun Lv, and Xipeng Qiu. Adalomo: Low-memory optimiza-
434 tion with adaptive learning rate, 2024.
- 435 [45] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena,
436 Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified
437 text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.
- 438 [46] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,
439 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to
440 solve math word problems, 2021. *Arxiv*, 2021.
- 441 [47] Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio
442 César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al.
443 Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 1(3):3, 2023.
- 444 [48] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei,
445 Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open
446 foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- 447 [49] Ashok Cutkosky and Harsh Mehta. Momentum improves normalized sgd, 2020.
- 448 [50] Kenji Kawaguchi. Deep learning without poor local minima. In D. Lee, M. Sugiyama,
449 U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing*
450 *Systems*, volume 29. Curran Associates, Inc., 2016.
- 451 [51] Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi.
452 MAWPS: A math word problem repository. In *Proceedings of the 2015 Conference of the*
453 *North American Chapter of the Association for Computational Linguistics: Human Language*
454 *Technologies*, pages 1152–1157, 2015.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [Yes]

Justification: we propose SUMO (Subspace-Aware Moment-Orthogonalization), an optimizer that employs exact singular value decomposition (SVD) for moment orthogonalization within a dynamically adapted low-dimensional subspace, enabling norm-inducing steepest descent optimization steps. By explicitly aligning optimization steps with the spectral characteristics of the loss landscape, SUMO effectively mitigates approximation errors associated with commonly used methods like Newton-Schulz orthogonalization approximation (claim. 3.3). We theoretically establish an upper bound on these approximation errors (claim. 3.2), proving their dependence on the condition numbers of moments, conditions we analytically demonstrate are encountered during LLM training. Furthermore, we both theoretically and empirically illustrate that exact orthogonalization via SVD substantially improves convergence rates while reducing overall complexity. Empirical evaluations confirm that SUMO accelerates convergence, enhances stability, improves performance, and reduces memory requirements by up to 20% compared to state-of-the-art methods. (Figure. 2, Table. 1)

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: Our methods work when the layers are reversible, and improves other methods especially when the moments are ill-conditioned.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: all assumptions of theoretical results are presented in the paper, and all proof are in the Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: Everything needed for reproducibility is presented in Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

- (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
- (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
- (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We provide experimental details in the main text and appendix. The code will be released after paper acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All training and test details are presented in Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All detailed in Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: All detailed in Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

Answer: [Yes]

Justification: The research conforms with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer:[NA]

Justification: The paper talks about optimizers. It does not have societal impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer:[NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We followed and gave proper credits for our use of data and models in Section 4.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human objects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human objects.

767 Guidelines:

768 • The answer NA means that the paper does not involve crowdsourcing nor research with

769 human subjects.

770 • Depending on the country in which research is conducted, IRB approval (or equivalent)

771 may be required for any human subjects research. If you obtained IRB approval, you

772 should clearly state this in the paper.

773 • We recognize that the procedures for this may vary significantly between institutions

774 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the

775 guidelines for their institution.

776 • For initial submissions, do not include any information that would break anonymity (if

777 applicable), such as the institution conducting the review.

778 **16. Declaration of LLM usage**

779 Question: Does the paper describe the usage of LLMs if it is an important, original, or

780 non-standard component of the core methods in this research? Note that if the LLM is used

781 only for writing, editing, or formatting purposes and does not impact the core methodology,

782 scientific rigorousness, or originality of the research, declaration is not required.

783 Answer: [NA]

784 Justification: The core method development in this research does not involve LLMs as any

785 important, original, or non-standard components.

786 Guidelines:

787 • The answer NA means that the core method development in this research does not

788 involve LLMs as any important, original, or non-standard components.

789 • Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>)

790 for what should or should not be described.

A Proofs of Section 3

In this section, we prove all the theorems and results of Section 3.

Lemma A.1 (Moment Becomes Low-Rank During Training). *Let $\mathbf{M}^{(t)} \in \mathbb{R}^{n \times m}$ denote the first moment of a reversible layer in a moment-based optimization algorithm, updated according to*

$$\mathbf{M}^{(t)} = \beta \mathbf{M}^{(t-1)} + \mathbf{G}^{(t)},$$

where $\mathbf{G}^{(t)}$ is the gradient matrix at iteration t . Let $\mathbf{M}^{(t)} = \mathbf{U}^{(t)} \mathbf{\Sigma}^{(t)} \mathbf{V}^{(t)\top}$ be the singular value decomposition (SVD) of $\mathbf{M}^{(t)}$, and define the rank- r orthogonal projection matrix as $\mathbf{P}^{(t)}(r) = \mathbf{U}^{(t)}[:, 1:r] \mathbf{U}^{(t)}[:, 1:r]^\top$. Then the relative error of the best rank-one approximation,

$$\kappa_M(t) \triangleq \frac{\|\mathbf{M}^{(t)} - \mathbf{P}^{(t)}(1) \mathbf{M}^{(t)}\|_F^2}{\|\mathbf{M}^{(t)}\|_F^2}, \quad (3)$$

satisfies $\kappa_M(t) \leq O(C^{-t})$ for some constant $C > 1$.

Proof of Lemma A.1. We aim to show that if the gradient $\mathbf{G}^{(t)}$ becomes approximately rank-one exponentially fast, then the exponentially weighted moving average of the gradients (i.e., the momentum $\mathbf{M}^{(t)}$) also exhibits exponential decay of higher-rank components.

Consider the singular value decomposition of the gradient $\mathbf{G}^{(t)} = \mathbf{U}^{(t)} \mathbf{\Sigma}^{(t)} \mathbf{V}^{(t)\top}$, at iteration t . For all natural numbers $r < m$, we define $\mathbf{H}^{(t)m \times r}(r) = \mathbf{U}[:, 1:r]$. To enhance notation clarity, denote $\mathbf{P}^{(t)}(r) = \mathbf{H}^{(t)}(r) \mathbf{H}^{(t)\top}(r)$, where $\mathbf{P}^{(t)}(r)$ represents an orthogonal projection matrix, satisfying the conditions $\mathbf{P}^{(t)\top}(r) \mathbf{P}^{(t)}(r) = \mathbf{P}^{(t)}(r)$, and $\mathbf{P}^{(t)}(r) = \mathbf{P}^{(t)\top}(r)$. Without compromising generality, it is assumed that at $t = 0$, the rank of $\mathbf{G}^{(0)}$ is characterized by $\text{rank}(\mathbf{G}^{(0)}) > r$. For reversible networks, it has been established in [1][Theorem 3.2] that the gradients assume the form $\mathbf{G}^{(t)} = \frac{1}{N} \sum_{i=1}^N (\mathbf{A}_i - \mathbf{B}_i \mathbf{W}^{(t)} \mathbf{C}_i)$, characterized by constant matrices $\{\mathbf{A}_i\}_i$ and positive semi-definite (PSD) matrices $\{\mathbf{B}_i, \mathbf{C}_i\}_i$, for $t \geq t_0$, where $t_0 \in \mathbb{N}$ holds. It is pertinent to recall that the vanilla weight update can be represented as $\mathbf{W}^{(t)} = \mathbf{W}^{(t-1)} + \eta \mathbf{G}^{(t-1)}$. Let $\mathbf{S} \triangleq \frac{1}{N} \sum_{i=1}^N \mathbf{C}_i \otimes \mathbf{B}_i$ and $\lambda_1 < \lambda_2$ denote its two smallest distinct eigenvalues. To substantiate our findings, we utilize several results and arguments presented in the proof of Lemma 3.3 in [1]. Specifically, consider $\mathbf{G}^{(t_0)}$ as the projection of $\mathbf{G}^{(t_0)}$ onto the minimal eigenspace \mathcal{V}_1 of \mathbf{S} corresponding to λ_1 . According to our assumption, the rank of $\mathbf{G}^{(t_0)}$ is L , and its singular value decomposition (SVD) is given by $\mathbf{G}^{(t_0)} = \sum_{l=1}^L c_l \mathbf{z}_l \mathbf{y}_l^\top$, where $\{\mathbf{z}_l\}_{l=1}^L$ and $\{\mathbf{y}_l\}_{l=1}^L$ are orthonormal unit vectors, and $\{c_l\}_{l=1}^L$ are the corresponding singular values. Therefore, as per Lemma 3.3 in [1], the gradient can be decomposed into,

$$\|\mathbf{G}^{(t)}\|_F^2 \leq (1 - \eta \lambda_2)^{2t} \|g_0^\perp\|_2^2 + (1 - \eta \lambda_1)^{2t} \|g_0^\parallel\|_2^2,$$

where g_0^\parallel is the projection of $\mathbf{G}^{(0)}$ onto the minimal eigenspace \mathcal{V}_1 of $\mathbf{S} = \frac{1}{N} \sum_{i=1}^N \mathbf{C}_i \otimes \mathbf{B}_i$, and g_0^\perp is orthogonal to \mathcal{V}_1 . Here, $\lambda_1 < \lambda_2$ are the smallest distinct eigenvalues of \mathbf{S} .

We now unroll the momentum update, $\mathbf{M}^{(t)} = \sum_{s=1}^t \beta^{t-s} \mathbf{G}^{(s)}$. Substitute the decomposition of $\mathbf{G}^{(s)}$,

$$\|\mathbf{M}^{(t)}\|_F^2 \leq \sum_{s=1}^t \beta^{t-s} \left[(1 - \eta \lambda_1)^s g_0^\parallel + (1 - \eta S)^s g_0^\perp \right] = \sum_{s=1}^t \beta^{t-s} (1 - \eta \lambda_1)^s g_0^\parallel + \sum_{s=1}^t \beta^{t-s} (1 - \eta S)^s g_0^\perp.$$

Let us define $a_t \triangleq \sum_{s=1}^t \beta^{t-s} (1 - \eta \lambda_1)^s$, $b_t \triangleq \sum_{s=1}^t \beta^{t-s} (1 - \eta S)^s g_0^\perp$, so that $\|\mathbf{M}^{(t)}\|_F^2 = a_t g_0^\parallel + b_t$. Now, compute the squared Frobenius norm:

$$\|\mathbf{M}^{(t)}\|_F^2 = \|a_t g_0^\parallel + b_t\|_F^2 = a_t^2 \|g_0^\parallel\|_F^2 + 2a_t \langle g_0^\parallel, b_t \rangle + \|b_t\|_F^2.$$

Since $g_0^\parallel \perp g_0^\perp$ and b_t lies in the span of g_0^\perp , we have $\langle g_0^\parallel, b_t \rangle = 0$, thus,

$$\|\mathbf{M}^{(t)}\|_F^2 = a_t^2 \|g_0\|_F^2 + \|b_t\|_F^2.$$

807 Likewise, the spectral norm $\|\mathbf{M}^{(t)}\|_2^2 \geq a_t^2 \|g_0\|_2^2$. Hence, the ratio

$$\kappa_m(t) = \frac{\|\mathbf{M}^{(t)} - \mathbf{P}^{(t)}(1)\mathbf{M}^{(t)}\|_F^2}{\|\mathbf{M}^{(t)}\|_F^2} \leq \frac{\|\mathbf{M}^{(t)}\|_F^2 - \|\mathbf{M}^{(t)}\|_2^2}{\|\mathbf{M}^{(t)}\|_F^2} \leq \frac{a_t^2 \|g_0\|_F^2 + \|b_t\|_F^2 - a_t^2 \|g_0\|_2^2}{a_t^2 \|g_0\|_F^2 + \|b_t\|_F^2}.$$

808 Using that $\|g_0\|_2^2 = \sigma_1^2$, and the decay bound $\|b_t\|_F^2 = O((\max\{\beta, 1 - \eta\lambda_2\})^{2t})$, while $a_t^2 =$
809 $\Omega((\max\{\beta, 1 - \eta\lambda_1\})^{2t})$, we conclude:

$$\kappa_m(t) \leq O\left(\left(\frac{\max\{\beta, 1 - \eta\lambda_2\}}{\max\{\beta, 1 - \eta\lambda_1\}}\right)^{2t}\right) = O(C^{-t}),$$

810 for some constant $C > 1$. □

811 Before proving Lemma 3.3, we shortly present the following two preliminary lemmas. To that end,
812 we present the following notations,

- 813 • $\mathbf{M}^{(t)}$ - The moment in iteration t . Its dimensions are $n \times m$, where $n < m$.
- 814 • $\|\cdot\|$ - The Frobenius norm: $\|\mathbf{A}\| = \|\mathbf{A}\|_F = \sqrt{\mathbf{A}\mathbf{A}^\top}$
- 815 • \mathcal{L}^* - A stationary point to which the loss \mathcal{L} converges.
- 816 • B - Batch size.
- 817 • For $\mathbf{A} \in \mathbb{R}^{m \times m}$ and $\mathbf{B} \in \mathbb{R}^{n \times n}$ we denote $\mathbf{A}\mathbf{I}_{m \times n}\mathbf{B}^\top$ by $\mathbf{A}\mathbf{B}^\top$ for convenience.

818 Additionally, we note that our proof is based on an equivalent but slightly modified formulation of
819 moment's update. Specifically, instead of using the standard formulation of the moment's update

$$\mathbf{M}^{(t+1)} = \beta\mathbf{M}^{(t)} + \mathbf{G}^{(t)},$$

820 we consider the convex combination,

$$\mathbf{M}^{(t+1)} = \beta\mathbf{M}^{(t)} + (1 - \beta)\mathbf{G}^{(t)}.$$

821 This alternative formulation simplifies the analysis, but equivalent. To show that, we point out that
822 we can choose an modified learning step $\eta^* = \frac{\eta}{1-\beta} > 0$ we get the same weight's updating step.

$$\begin{aligned} & \eta^* \text{Orth}\left(\beta\mathbf{M}^{(t)} + (1 - \beta)\mathbf{G}^{(t)}\right) \\ &= \text{Orth}\left(\eta^* \beta\mathbf{M}^{(t)} + \eta^*(1 - \beta)\mathbf{G}^{(t)}\right) \\ &= \text{Orth}\left(\eta \frac{\beta}{1 - \beta}\mathbf{M}^{(t)} + \eta\mathbf{G}^{(t)}\right) \\ &= \eta \text{Orth}\left(\frac{\beta}{1 - \beta}\mathbf{M}^{(t)} + \mathbf{G}^{(t)}\right). \end{aligned}$$

823 Where Orth is the SVD orthogonalization step. Obviously, $\beta > 0$ could be chosen in a way that
824 $\frac{\beta}{1 - \beta}\mathbf{M}^{(t)}$ would result in any required positive real number.

825 We assume the following 4 assumptions throughout our proofs:

- 826 (A1) The gradient $\nabla\mathcal{L}(\mathbf{W})$ is L -Lipschitz continuous.
- 827 (A2) $\nabla\mathcal{L}(\mathbf{W}, \xi)$ is an unbiased estimator of $\nabla\mathcal{L}(\mathbf{W})$ where $\mathcal{L}(\mathbf{W}, \xi)$ is the gradient of $\mathcal{L}(\mathbf{W})$
828 when taking a single training sample ξ .
- 829 (A3) $\mathbb{E}\|\nabla\mathcal{L}(\mathbf{W}, \xi) - \nabla\mathcal{L}(\mathbf{W})\| \leq \sigma^2$.

830 (A4) There exists $\delta > 0$ such that $\|\mathcal{E}_5^{(t)}\| \leq \delta \|\mathbf{U}^{(t)} \mathbf{V}^{(t)\top}\| = \delta \sqrt{m}$ for all t .

831 **Lemma A.2** (Descent Lemma with Newton-Schulz Approximation Error). *Consider the Muon*
 832 *optimizer update defined by*

$$\begin{aligned}\mathbf{M}^{(t)} &\leftarrow \beta \mathbf{M}^{(t-1)} + (1 - \beta) \mathbf{G}^{(t)}, \\ \mathbf{O}^{(t)} &\leftarrow \mathbf{U}^{(t)} \mathbf{V}^{(t)\top} + \mathcal{E}_5^{(t)}, \quad (\text{Newton-Schulz 5 iteration approximation}), \\ \mathbf{W}^{(t+1)} &\leftarrow \mathbf{W}^{(t)} - \eta_t \mathbf{O}^{(t)},\end{aligned}$$

833 where $\mathbf{M}^{(t)} = \mathbf{U}^{(t)} \mathbf{S}^{(t)} \mathbf{V}^{(t)\top}$ is the singular value decomposition of $\mathbf{M}^{(t)}$, and $\mathcal{E}_5^{(t)}$ represents
 834 the Newton-Schulz (5 iterations) approximation error. Additionally, assume (A1) - (A4). Then the
 835 following holds:

$$\begin{aligned}\mathcal{L}(\mathbf{W}^{(t+1)}) &\leq \\ \mathcal{L}(\mathbf{W}^{(t)}) - \left(\frac{\eta_t}{4} - \eta_t \sqrt{m} \delta\right) \|\nabla \mathcal{L}(\mathbf{W}^{(t)})\| &+ \eta_t \frac{5}{2} \|\nabla \mathcal{L}(\mathbf{W}^{(t)}) - \mathbf{M}^{(t)}\| + \frac{\eta_t^2 m L}{2} + \eta_t^2 m L \delta + \frac{\eta_t^2 L m \delta^2}{2}\end{aligned}$$

836 *Proof.* Since \mathcal{L} is L-lipschitz function, the descent lemma holds. Thus we have

$$\begin{aligned}\mathcal{L}(\mathbf{W}^{(t+1)}) &\leq \mathcal{L}(\mathbf{W}^{(t)}) + \langle \nabla \mathcal{L}(\mathbf{W}^{(t)}), \mathbf{W}^{(t+1)} - \mathbf{W}^{(t)} \rangle + \frac{L}{2} \|\mathbf{W}^{(t+1)} - \mathbf{W}^{(t)}\|^2 \\ &= \mathcal{L}(\mathbf{W}^{(t)}) - \eta_t \langle \nabla \mathcal{L}(\mathbf{W}^{(t)}), \mathbf{U}^{(t)} \mathbf{V}^{(t)\top} + \mathcal{E}_5^{(t)} \rangle + \frac{L \eta_t^2}{2} \|\mathbf{U}^{(t)} \mathbf{V}^{(t)\top} + \mathcal{E}_5^{(t)}\|^2 \\ &= \mathcal{L}(\mathbf{W}^{(t)}) - \eta_t \langle \nabla \mathcal{L}(\mathbf{W}^{(t)}), \mathbf{U}^{(t)} \mathbf{V}^{(t)\top} \rangle - \eta_t \langle \nabla \mathcal{L}(\mathbf{W}^{(t)}), \mathcal{E}_5^{(t)} \rangle \\ &\quad + \frac{L \eta_t^2}{2} \left(n + 2 \langle \mathbf{U}^{(t)} \mathbf{V}^{(t)\top}, \mathcal{E}_5^{(t)} \rangle + \|\mathcal{E}_5^{(t)}\|^2 \right) \\ &\stackrel{(*)}{\leq} \mathcal{L}(\mathbf{W}^{(t)}) - \frac{\eta_t}{4} \|\nabla \mathcal{L}(\mathbf{W}^{(t)})\| + \eta_t \frac{5}{2} \|\nabla \mathcal{L}(\mathbf{W}^{(t)}) - \mathbf{M}^{(t)}\| + \frac{\eta_t^2 m L}{2} \\ &\quad + \eta_t \|\nabla \mathcal{L}(\mathbf{W}^{(t)})\| \|\mathcal{E}_5^{(t)}\| + L \eta_t^2 \sqrt{m} \|\mathcal{E}_5^{(t)}\| + \frac{L \eta_t^2}{2} \|\mathcal{E}_5^{(t)}\|^2 \\ &\leq \mathcal{L}(\mathbf{W}^{(t)}) - \frac{\eta_t}{4} \|\nabla \mathcal{L}(\mathbf{W}^{(t)})\| + \eta_t \frac{5}{2} \|\nabla \mathcal{L}(\mathbf{W}^{(t)}) - \mathbf{M}^{(t)}\| + \frac{\eta_t^2 m L}{2} \\ &\quad + \eta_t \delta \sqrt{m} \|\nabla \mathcal{L}(\mathbf{W}^{(t)})\| + L \eta_t^2 \sqrt{m} \delta \sqrt{m} + \frac{L \eta_t^2}{2} \delta^2 n \\ &= \mathcal{L}(\mathbf{W}^{(t)}) - \left(\frac{\eta_t}{4} - \eta_t \sqrt{m} \delta\right) \|\nabla \mathcal{L}(\mathbf{W}^{(t)})\| + \eta_t \frac{5}{2} \|\nabla \mathcal{L}(\mathbf{W}^{(t)}) - \mathbf{M}^{(t)}\| + \frac{\eta_t^2 m L}{2} + \\ &\quad + \eta_t^2 m L \delta + \frac{\eta_t^2 L m \delta^2}{2}\end{aligned}$$

837 Where in (*) we used [49], equation 2.8. □

838 **Lemma A.3.** For constant $\eta_t = \eta > 0$, the following holds

$$\begin{aligned}\frac{\eta - 4\eta\sqrt{m}\delta}{4} \sum_{t=1}^T \|\nabla \mathcal{L}(\mathbf{W}^{(t)})\| &\leq \\ \mathcal{L}(\mathbf{W}^{(1)}) - \mathcal{L}^* + \eta \frac{5}{2} \sum_{t=1}^T \|\nabla \mathcal{L}(\mathbf{W}^{(t)}) - \mathbf{M}^{(t)}\| &+ \frac{\eta^2 m L T}{2} + T \eta^2 m L \delta + \frac{T \eta^2 L m \delta^2}{2}.\end{aligned}$$

839 *Proof.* Using Lemma A.2, isolating $\|\nabla \mathcal{L}(\mathbf{W}^{(t)})\|$ and summing over all steps

$$\begin{aligned}\frac{\eta - 4\eta\sqrt{m}\delta}{4} \sum_{t=1}^T \|\nabla \mathcal{L}(\mathbf{W}^{(t)})\| &\leq \\ \mathcal{L}(\mathbf{W}^{(1)}) - \mathcal{L}^* + \eta \frac{5}{2} \sum_{t=1}^T \|\nabla \mathcal{L}(\mathbf{W}^{(t)}) - \mathbf{M}^{(t)}\| &+ \frac{\eta^2 m L T}{2} + T \eta^2 m L \delta + \frac{T \eta^2 L m \delta^2}{2}\end{aligned}$$

840 □

841 **Lemma 3.3** (Exact convergence rate of Muon). *Consider the Muon optimizer update defined by*

$$\begin{aligned}\mathbf{M}^{(t)} &\leftarrow \beta \mathbf{M}^{(t-1)} + (1 - \beta) \mathbf{G}_t \\ \mathbf{O}^{(t)} &\leftarrow \mathbf{U}^{(t)} \mathbf{V}^{(t)\top} + \mathcal{E}_i^{(t)}, \quad (i \text{ iterations Newton-Schulz approximation}) \\ \mathbf{W}^{(t+1)} &\leftarrow \mathbf{W}^{(t)} - \eta_t \mathbf{O}^{(t)},\end{aligned}$$

where $\mathbf{M}^{(t)} = \mathbf{U}^{(t)} \mathbf{S}^{(t)} \mathbf{V}^{(t)\top}$ denotes the singular value decomposition of $\mathbf{M}^{(t)}$, and $\mathcal{E}_i^{(t)}$ represents the Newton-Schulz approximation error after i iterations. Assuming A(1) - A(4), If we take $\beta = 1 - \alpha$ with $\alpha = \min(\frac{\sqrt{RL}}{\sigma\sqrt{T}}, 1)$, $\eta_t = \eta = \frac{\sqrt{4R}}{\sqrt{(10/(1-\beta)+2m+4m\delta+2m\delta^2)TL}}$, and $B = 1$ (batch free convergence) then $\frac{1}{T} \sum_{t=1}^T \mathbb{E} [\|\nabla \mathcal{L}(\mathbf{W}^{(t)})\|]$ is bounded by

$$\mathcal{O} \left(\left[\frac{\sqrt{RLn(2+4\delta+2\delta^2)}}{\sqrt{T}} + \frac{\sigma^2}{\sqrt{RLT}} + \frac{\sigma(RL)^{1/4} + \sqrt{\sigma}(RL)^{1/4}}{T^{1/4}} \right] \frac{1}{1-4\sqrt{m\delta}} \right),$$

842 where $R = \mathcal{L}(\mathbf{W}^{(0)}) - \mathcal{L}^*$. If we take β as an arbitrary constant, we have to take $B = T$, and we
843 have,

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \|\nabla \mathcal{L}(\mathbf{W}^{(t)})\| \leq \mathcal{O} \left(\left[\frac{\sqrt{RLn(2+4\delta+2\delta^2)}}{\sqrt{T}} + \frac{\sqrt{RL}}{\sqrt{T}} + \frac{\sigma}{T^{3/2}} + \frac{\sigma}{\sqrt{T}} \right] \frac{1}{1-4\sqrt{m\delta}} \right).$$

844 *Proof of Lemma 3.3.* The proof follows [49]. Using the same notations as [49], we denote
845 $\hat{\gamma}^{(t)} = \mathbf{M}^{(t)} - \nabla \mathcal{L}(\mathbf{W}^{(t)})$, $\gamma^{(t)} = \mathbf{G}^{(t)} - \nabla \mathcal{L}(\mathbf{W}^{(t)})$ and $S(\mathbf{X}, \mathbf{Y}) = \nabla \mathcal{L}(\mathbf{X}) - \nabla \mathcal{L}(\mathbf{Y})$. Note
846 that we have the following

- 847 • $\mathbb{E}[\gamma^{(t)}] = 0$ from A(2).
- 848 • $\mathbb{E}[\|\gamma^{(t)}\|^2] \leq \frac{\sigma^2}{m}$ from A(3).
- 849 • $\mathbb{E}[\langle \gamma^{(i)}, \gamma^{(j)} \rangle] = 0$, $\forall i \neq j$ since $\gamma^{(i)}$ and $\gamma^{(j)}$ are independent.
- 850 • $\|S(\mathbf{X}, \mathbf{Y})\| \leq L\|\mathbf{X} - \mathbf{Y}\|$ from A(1).

851 Now following the update in (2), we get

$$\begin{aligned}\hat{\gamma}^{(t+1)} &= \beta \hat{\gamma}^{(t)} + (1 - \beta) \gamma^{(t)} + S(\mathbf{X}^{(t)}, \mathbf{X}^{(t+1)}) \\ &= \beta^t \hat{\gamma}^{(1)} + (1 - \beta) \sum_{\tau=0}^{t-1} \beta^\tau \gamma^{(t-\tau)} + \sum_{\tau=0}^{t-1} \beta^\tau S(\mathbf{X}^{(t-\tau)}, \mathbf{X}^{(t+1-\tau)}),\end{aligned}$$

852 therefore

$$\|\hat{\gamma}^{(t+1)}\| \leq \beta^t \|\hat{\gamma}^{(1)}\| + (1 - \beta) \left\| \sum_{\tau=0}^{t-1} \beta^\tau \gamma^{(t-\tau)} \right\| + \eta L \sum_{\tau=0}^{t-1} \beta^\tau.$$

853 Taking expectation we get (using the fact that $\hat{\delta}_1 = \delta_1$):

$$\begin{aligned}\mathbb{E} \|\hat{\gamma}^{(t+1)}\| &\leq \beta^t \frac{\sigma}{m} + (1 - \beta) \sqrt{\sum_{\tau=0}^{t-1} \beta^{2\tau} \frac{\sigma^2}{B}} + \eta L \sum_{\tau=0}^{t-1} \beta^\tau \\ &\leq \frac{\sigma}{m} \beta^t + \frac{\sigma}{m} \frac{1 - \beta}{\sqrt{1 - \beta^2}} + \eta L \frac{1}{1 - \beta} \\ &\leq \frac{\sigma}{m} \beta^t + \frac{\sigma}{m} \sqrt{1 - \beta} + \eta L \frac{1}{1 - \beta}.\end{aligned}$$

All in all, we get

$$\sum_{t=1}^T \mathbb{E} [\|\hat{\gamma}^{(t+1)}\|] \leq \frac{\sigma}{(1 - \beta)B} + T \sqrt{1 - \beta} \frac{\sigma}{m} + \frac{T \eta L}{1 - \beta}.$$

854 Using Lemma A.3, we get

$$\begin{aligned} & \frac{\eta_t - 4L\eta_t\sqrt{m}\delta}{4} \sum_{t=1}^T \|\nabla\mathcal{L}(\mathbf{W}^{(t)})\| \leq \\ & \mathcal{L}(\mathbf{W}^{(1)}) - \mathcal{L}^* + \eta_t \frac{5}{2} \sum_{t=1}^T \|\nabla\mathcal{L}(\mathbf{W}^{(t)}) - \mathbf{M}^{(t)}\| + \frac{\eta^2 m L T}{2} + T\eta^2 m L \delta + \frac{T\eta^2 L m \delta^2}{2}. \end{aligned}$$

855 Dividing both sides by $\frac{\eta - 4\eta L\sqrt{m}\delta}{4}$ we get

$$\begin{aligned} & \sum_{t=1}^T \|\nabla\mathcal{L}(\mathbf{W}^{(t)})\| \leq \\ & \left[\frac{4(\mathcal{L}(\mathbf{W}^{(1)}) - \mathcal{L}^*)}{\eta} + 10 \sum_{t=1}^T \|\nabla\mathcal{L}(\mathbf{W}^{(t)}) - \mathbf{M}^{(t)}\| + 2\eta m L T + 4T\eta m L \delta + 2T\eta L m \delta^2 \right] \cdot \frac{1}{1 - 4\sqrt{m}\delta} \\ & \leq \left[\frac{4R}{\eta} + 10 \frac{\sigma}{(1-\beta)m} + 10T\sqrt{1-\beta} \frac{\sigma}{m} + 10 \frac{T\eta L}{1-\beta} + 2\eta m L T + 4T\eta m L \delta + 2T\eta L m \delta^2 \right] \cdot \frac{1}{1 - 4\sqrt{m}\delta} \end{aligned}$$

856 By taking $\eta = \sqrt{\frac{4R}{(10/(1-\beta)+2m+4m\delta+2m\delta^2)TL}}$ we get

$$\begin{aligned} & \sum_{t=1}^T \|\nabla\mathcal{L}(\mathbf{W}^{(t)})\| \\ & \leq \left[4\sqrt{RTL(10/(1-\beta)+2m+4m\delta+2m\delta^2)} + \frac{10\sigma}{(1-\beta)m} + 10T\sqrt{1-\beta} \frac{\sigma}{(1-\beta)m} \right] \frac{1}{1 - 4\sqrt{m}\delta}. \end{aligned}$$

857 Now we have two types of parameter choice. If we take $B = 1$ (batch size free), we need to take $1 - \beta =$

858 $\min(1, \frac{\sqrt{RL}}{\sigma\sqrt{T}})$ so that we have

$$\begin{aligned} & \sum_{t=1}^T \|\nabla\mathcal{L}(\mathbf{W}^{(t)})\| \\ & \leq \left[2\sqrt{RTL(2m+4m\delta+2m\delta^2)} + 2\sqrt{10} \cdot \sigma \cdot (RL)^{1/4} T^{3/4} + 10\sigma^2 \sqrt{\frac{T}{RL}} + 10\sqrt{\sigma}(RL)^{1/4} T^{3/4} \right] \frac{1}{1 - 4\sqrt{m}\delta}, \end{aligned}$$

thus

$$\frac{1}{T} \sum_{t=1}^T E \left[\|\nabla\mathcal{L}(\mathbf{W}^{(t)})\| \right] \leq \mathcal{O} \left(\left[\frac{\sqrt{RLn(2+4\delta+2\delta^2)}}{\sqrt{T}} + \sigma \cdot \frac{(RL)^{1/4}}{T^{1/4}} + \frac{\sigma^2}{\sqrt{RLT}} + \frac{\sqrt{\sigma}(RL)^{1/4}}{T^{1/4}} \right] \frac{1}{1 - 4\sqrt{m}\delta} \right).$$

859 If we take β as an arbitrary constant in $(0, 1)$, then we will need to take $B = T$, so that

$$\frac{1}{T} \sum_{t=1}^T \|\nabla\mathcal{L}(\mathbf{W}^{(t)})\| \leq \mathcal{O} \left(\left[\frac{\sqrt{RLn(2+4\delta+2\delta^2)}}{\sqrt{T}} + \frac{\sqrt{RL}}{\sqrt{T}} + \frac{\sigma}{T^{3/2}} + \frac{\sigma}{\sqrt{T}} \right] \frac{1}{1 - 4\sqrt{m}\delta} \right).$$

860

□

861 **Lemma 3.2** (Orthogonalization error \mathcal{E}_i) For a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, let σ_1 be the largest singular
862 value of $\mathbf{A}\mathbf{A}^\top$ and σ_m be the smallest (without the loss of generality, assume $m \leq n$). Let $r \leq m$ be
863 the largest index where $\sigma_r > \sigma_{r+1} = \dots = \sigma_m \geq 0$. Let $\kappa = \frac{\sigma_1}{\sigma_m}$ by the condition number of $\mathbf{A}\mathbf{A}^\top$.
864 Denote \mathcal{E}_i the error of Newton-Schultz after i iterations. Then we have

$$\|\mathcal{E}_i\|_F \leq \sqrt{r} \cdot \left(1 - \frac{1}{\kappa} \right)^{2^i}. \quad (4)$$

865 *Proof of Lemma 3.2.* We denote $\mathbf{B} = \mathbf{A}\mathbf{A}^\top$, \mathbf{X}_k the result after k Newton-Schultz iterations, $\mathbf{X}_0 =$
 866 $\frac{\mathbf{B}}{\|\mathbf{B}\|_2}$ and $\mathbf{O} = \mathbf{U}\mathbf{V}^\top$ where $\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ is the SVD decomposition with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_m$ the
 867 singular values.

It is known that Newton-Schultz converges quadratically, so we have

$$\|\mathcal{E}_k\|_2 = \|\mathbf{X}_k - \mathbf{O}\|_2 \leq \|\mathbf{X}_0 - \mathbf{O}\|_2^{2^k}$$

We now bound $\|\mathbf{X}_0 - \mathbf{O}\|_2$. We know that $\mathbf{X}_0 = \frac{\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top}{\|\mathbf{B}\|_2} = \frac{\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top}{\sigma_1}$

$$\|\mathbf{X}_0 - \mathbf{O}\|_2 = \left\| \frac{\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top}{\sigma_1} - \mathbf{U}\mathbf{V}^\top \right\|_2 = \left\| \mathbf{U} \left(\frac{\mathbf{\Sigma}}{\sigma_1} - \mathbf{I} \right) \mathbf{V}^\top \right\|_2 = \left\| \frac{\mathbf{\Sigma}}{\sigma_1} - \mathbf{I} \right\|_2$$

Where last equality is due to the fact that $\|\cdot\|_2$ is unitary invariant. The matrix $\frac{\mathbf{\Sigma}}{\sigma_1} - \mathbf{I}$ is diagonal with values $\frac{\sigma_i}{\sigma_1} - 1$ on the diagonal. From that observation we get that

$$\left\| \frac{\mathbf{\Sigma}}{\sigma_1} - \mathbf{I} \right\|_2 = \max_i \left| \frac{\sigma_i}{\sigma_1} - 1 \right| = \max_i \left(1 - \frac{\sigma_i}{\sigma_1} \right) = 1 - \frac{\sigma_m}{\sigma_1} = 1 - \frac{1}{\kappa}$$

For the Frobenius norm, we get a similar analysis. $\|\mathbf{X}_0 - \mathbf{O}\|_F = \left\| \frac{\mathbf{\Sigma}}{\sigma_1} - \mathbf{I} \right\|_F$ since $\|\cdot\|_F$ is unitary invariant, so we just need to calculate $\left\| \frac{\mathbf{\Sigma}}{\sigma_1} - \mathbf{I} \right\|_F$. It is known that $\left\| \frac{\mathbf{\Sigma}}{\sigma_1} - \mathbf{I} \right\|_F \leq \sqrt{r} \left\| \frac{\mathbf{\Sigma}}{\sigma_1} - \mathbf{I} \right\|_2$ so all in all we have

$$\left\| \frac{\mathbf{\Sigma}}{\sigma_1} - \mathbf{I} \right\|_F \leq \sqrt{r} \left\| \frac{\mathbf{\Sigma}}{\sigma_1} - \mathbf{I} \right\|_2 = \sqrt{r} \left(1 - \frac{1}{\kappa} \right).$$

868

□

869 **Theorem 3.8** (Convergence of SUMO) *For a loss function \mathcal{L} , and given architecture Φ , suppose*
 870 *that the compositions of $f \equiv \mathcal{L}(\Phi(\cdot))$ is β -smooth non-convex function that is bounded by some*
 871 *$M \in \mathbb{R}_+$. Let $\mathbf{G}_j^{(t)}$ denote the gradient matrix w.r.t. the j -th reversible layer $\mathbf{W}_j^{(t)}$, at time $t \in \mathbb{N}$, for*
 872 *all $j \in [L]$ and $t \in \mathbb{N}$, and $T_\ell, \ell \in \mathbb{N}$ times are set by a convergence criterion (that is, $\|\hat{\mathbf{G}}_j^{(T_\ell)}\| \leq \varsigma_\ell$).*
 873 *Then, there exist $C \in \mathbb{R}_+$ and N such that for all $T_N > \frac{C}{\varepsilon^2}$, and $\frac{1}{T_N} \sum_{i=0}^{N-1} \sum_{t=T_i}^{T_{i+1}-1} \left\| \mathbf{G}_j^{(t)} \right\|_F^2 \leq \varepsilon$.*
 874 *Namely, Algorithm 1 achieves an ε -critical point,³ i.e., $\left\| \mathbf{G}_j^{(t)} \right\|_F^2 \leq \varepsilon$, for some $t \in \mathbb{N}$, and any*
 875 *$j \in [L]$.*

876 *Proof of Theorem 3.8.* for any layer $j \in [L]$; in the following, for simplicity of notation, we ignore
 877 the index j and use $\mathbf{G}^{(t)}$ instead. By Lemma A.4, the low-rank optimization block 1 in Algorithm 1
 878 is guaranteed to converge; we denote by $T_\ell \in \mathbb{N}$ the time index t at which we exit block 3 for the ℓ th
 879 time (i.e., $\|\hat{\mathbf{G}}^{(T_\ell)}\| \leq \varsigma_2$), for $\ell \in \mathbb{N}$. Furthermore, we recall that $\mathbf{G}_j^{(t)} \triangleq \nabla_{\mathbf{W}_j} f(\boldsymbol{\theta}^{(t)})$; when clear
 880 from the context, we omit j from \mathbf{W}_j , and use instead $\nabla_{\mathbf{W}_j} f(\boldsymbol{\theta}^{(t)}) = \nabla f(\mathbf{W}^{(t)})$. Consider the
 881 SVD decomposition of the gradient $\nabla_{\mathbf{W}_j} f(\boldsymbol{\theta}_{T_i}) = \mathbf{U}^{(T_i)} \mathbf{\Sigma}^{(T_i)} \mathbf{V}^{(T_i)\top}$. For $t \in [T_i, T_{i+1} - 1]$, we
 882 define the projected gradient as $\hat{\mathbf{G}}^{(t)} \triangleq \mathbf{P}^{(T_i)}(r) \mathbf{G}^{(t)}$, where $\mathbf{P}^{(T_i)}(r) = \mathbf{U}^{(T_i)}[:, :r]^\top$, using the
 883 exact truncated-SVD calculation (in Block 1). For simplicity, we refer to $\mathbf{P}^{(T_i)}(r)$ as $\mathbf{P}^{(T_i)}$. Next, let
 884 $h_t \triangleq f(\mathbf{W}^{(t)}) - f(\mathbf{W}^{(T_{i+1})})$, and η_t denote the learning rate. Then,

$$\begin{aligned} h_{t+1} &= f(\mathbf{W}^{(t+1)}) - f(\mathbf{W}^{(T_{i+1})}) \\ &= f(\mathbf{W}^{(t)} - \eta_t (\mathbf{W}^{(t+1)} - \mathbf{W}^{(t)})) - f(\mathbf{W}^{(T_{i+1})}) \\ &\stackrel{(1)}{\leq} f(\mathbf{W}^{(t)}) - f(\mathbf{W}^{(T_{i+1})}) - \eta_t \text{vec}(\hat{\mathbf{O}}^{(t)})^\top \text{vec}(\mathbf{G}^{(t)}) + \eta_t^2 \frac{\beta}{2} \|\hat{\mathbf{O}}^{(t)}\|_F^2 \end{aligned}$$

³Also known as ε -stationary, see, e.g., [12].

$$\begin{aligned}
&= f(\mathbf{W}^{(t)}) - f(\mathbf{W}^{(\tau_{i+1})}) - \frac{\eta_t}{4} \|\mathbf{G}^{(t)}\| + \eta_t \frac{5}{2} \|\mathbf{G}^{(t)} - \mathbf{M}^{(t)}\| + \frac{\eta_t^2 m \beta}{2} \\
&= f(\mathbf{W}^{(t)}) - f(\mathbf{W}^{(\tau_{i+1})}) - \eta_t \text{vec}(\hat{\mathbf{G}}^{(t)})^\top \text{vec}(\mathbf{G}^{(t)}) + \eta_t^2 \frac{\beta}{2} \|\hat{\mathbf{G}}^{(t)}\|_F^2
\end{aligned} \tag{5}$$

By Lemma A.4,

$$= h_t - \frac{4}{\eta T} \left(f(\mathbf{W}^{(1)}) - f^* \right) + 10 \frac{\sigma}{\sqrt{m}} + 2\eta m L. \tag{6}$$

Summing (6) over $t = \tau_i$ to $\tau_{i+1} - 1$ and using constant step size $\eta_t = \eta = \frac{\sqrt{4R}}{\sqrt{(10/(1-\beta)+2m+4m\delta+2m\delta^2)TL}}$, we get:

$$\sum_{t=\tau_i}^{\tau_{i+1}-1} \mathbb{E} \|\hat{\mathbf{G}}^{(t)}\|_F^2 \leq \frac{2(h_{\tau_i} - h_{\tau_{i+1}})}{\eta} \leq \frac{2(h_{\tau_i} - h_{\tau_{i+1}})(10/(1-\beta) + 2m + 4m\delta + 2m\delta^2)\sqrt{TL}}{\sqrt{4R}}. \tag{7}$$

Summing over $i = 0$ to $N - 1$:

$$\frac{1}{\tau_N} \sum_{i=0}^{N-1} \sum_{t=\tau_i}^{\tau_{i+1}-1} \mathbb{E} \|\hat{\mathbf{G}}^{(t)}\|_F^2 \leq \frac{M\sqrt{TL}}{\sqrt{4R}}. \tag{8}$$

Let \mathbf{Q}_{τ_i} be the projection onto an informative subspace such that:

$$\|\hat{\mathbf{G}}^{(\tau_i)} - \mathbf{Q}_{\tau_i} \hat{\mathbf{G}}^{(\tau_i)}\|_F^2 \leq \alpha \|\hat{\mathbf{G}}^{(\tau_i)}\|_F^2. \tag{9}$$

Then,

$$\|\mathbf{Q}_{\tau_i}^\perp \hat{\mathbf{G}}^{(\tau_i)}\|_F^2 \leq \frac{\alpha}{1-\alpha} \|\mathbf{Q}_{\tau_i} \hat{\mathbf{G}}^{(\tau_i)}\|_F^2. \tag{10}$$

From Lemma B.3 in [1], under $\eta \leq \frac{2}{\lambda_{\max}}$, we get:

$$\mathbb{E} \|\hat{\mathbf{G}}^{(t)}\|_F^2 \leq \mathbb{E} \|\hat{\mathbf{G}}^{(\tau_i)}\|_F^2, \quad \forall t \in [\tau_i, \tau_{i+1}). \tag{11}$$

Hence,

$$\frac{1}{\tau_N} \sum_{i=0}^{N-1} \sum_{t=\tau_i}^{\tau_{i+1}-1} \|\hat{\mathbf{G}}^{(t)}\|_F^2 \leq \frac{1}{\tau_N} \sum_{i=0}^{N-1} (\tau_{i+1} - \tau_i) \|\hat{\mathbf{G}}^{(\tau_i)}\|_F^2 \tag{12}$$

$$\leq \frac{1}{(1-\alpha)\tau_N} \sum_{i=0}^{N-1} (\tau_{i+1} - \tau_i) \|\mathbf{Q}_{\tau_i} \hat{\mathbf{G}}^{(\tau_i)}\|_F^2 \tag{13}$$

$$\leq \frac{M}{(1-\alpha)\beta\sqrt{\tau_N}}. \tag{14}$$

Finally, for any $\varepsilon_1, \varepsilon_2 > 0$, if we choose $\tau_N > \frac{M^2}{(1-\alpha)^2\varepsilon_1^2}$ and $\rho_N < \varepsilon_2 \cdot \frac{1-\alpha}{\beta^2}$, then for $\varepsilon \geq \varepsilon_1 + \varepsilon_2$,

$$\min_{0 \leq t \leq \tau_N} \|\hat{\mathbf{G}}^{(t)}\|_F^2 \leq \varepsilon. \tag{15}$$

This concludes that Algorithm 1 achieves an ε -critical point.

□

In the following, we provide the auxiliary lemma that is used in the proof of Theorem 3.8.

Lemma A.4 (Convergence of the Inner Fixed Low-Rank Optimization). *Consider the same setting and assumptions as in Theorem 3.8. Then, the second time $t = \tau_\ell \in \mathbb{N}$ in which Algorithm 1 enters Block 1 (where it updates the projection matrix) happens for a finite $\ell \in \mathbb{N}$.*

900 *Proof.* By the L -smoothness of f , we have,

$$f(\mathbf{W}^{(t+1)}) \leq f(\mathbf{W}^{(t)}) + \langle \mathbf{G}^{(t)}, \mathbf{W}^{(t+1)} - \mathbf{W}^{(t)} \rangle + \frac{L}{2} \|\mathbf{W}^{(t+1)} - \mathbf{W}^{(t)}\|_F^2.$$

901 Substituting the update rule $\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta_t \mathbf{O}^{(t)}$ gives,

$$f(\mathbf{W}^{(t+1)}) \leq f(\mathbf{W}^{(t)}) - \eta_t \langle \mathbf{G}^{(t)}, \mathbf{O}^{(t)} \rangle + \frac{L\eta_t^2}{2} \|\mathbf{O}^{(t)}\|_F^2.$$

902 Since, $\hat{\mathbf{G}}^{(t)} = \mathbf{P}^{(t)}(r)^\top \mathbf{G}^{(t)}$, then,

$$\langle \mathbf{G}^{(t)}, \mathbf{O}^{(t)} \rangle = \langle \hat{\mathbf{G}}^{(t)}, \mathbf{O}^{(t)} \rangle,$$

903 since $\mathbf{O}^{(t)} \in \text{range}(\mathbf{P}^{(t)}(r)^\top)$.

904 Thus, by [39] equation (2.8) we have

$$-\langle \hat{\mathbf{G}}^{(t)}, \mathbf{O}^{(t)} \rangle = -\langle \mathbf{G}^{(t)}, \mathbf{O}^{(t)} \rangle \leq -\frac{1}{4} \|\mathbf{G}^{(t)}\| + \frac{5}{2} \|\mathbf{G}^{(t)} - \mathbf{M}^{(t)}\|$$

905 Now we bound $\|\mathbf{G}^{(t)} - \mathbf{M}^{(t)}\|$,

$$\begin{aligned} \|\mathbf{G}^{(t)} - \mathbf{M}^{(t)}\| &= \|\mathbf{G}^{(t)} - \nabla f(\mathbf{W}^{(t)}) + \nabla f(\mathbf{W}^{(t)}) - \mathbf{M}^{(t)}\| \\ &\leq \|\mathbf{G}^{(t)} - \nabla f(\mathbf{W}^{(t)})\| + \|\nabla f(\mathbf{W}^{(t)}) - \mathbf{M}^{(t)}\| \\ &\stackrel{(1)}{\leq} \frac{\sigma}{\sqrt{m}}, \end{aligned}$$

906 where (1) is from assumptions A(2) and A(3). Therefore,

$$-\langle \hat{\mathbf{G}}^{(t)}, \mathbf{O}^{(t)} \rangle \leq -\frac{1}{4} \|\hat{\mathbf{G}}^{(t)}\| + \frac{5}{2} \|\mathbf{G}^{(t)} - \mathbf{M}^{(t)}\| \leq -\frac{1}{4} \|\hat{\mathbf{G}}^{(t)}\| + \frac{\sigma}{\sqrt{m}}.$$

This all comes down to the end,

$$f(\mathbf{W}^{(t+1)}) \leq f(\mathbf{W}^{(t)}) - \frac{\eta_t}{4} \|\hat{\mathbf{G}}^{(t)}\| + \frac{5}{2} \frac{\eta_t \sigma}{\sqrt{m}} + \frac{\eta_t^2 n L}{2},$$

907 hence for constant step size $\eta_t = \eta$, this simplifies to

$$f(\mathbf{W}^{(T+1)}) \leq f(\mathbf{W}^{(1)}) - \frac{\eta}{4} \sum_{t=1}^T \|\hat{\mathbf{G}}^{(t)}\| + \frac{5}{2} \frac{\eta \sigma T}{\sqrt{m}} + \frac{\eta^2 n L T}{2},$$

908 and finally,

$$\frac{1}{T} \sum_{t=1}^T \|\hat{\mathbf{G}}^{(t)}\| \leq \frac{4}{\eta T} (f(\mathbf{W}^{(1)}) - f^*) + 10 \frac{\sigma}{\sqrt{m}} + 2\eta n L \quad (16)$$

$$= \frac{4}{\eta T} M + 10 \frac{\sigma}{\sqrt{m}} + 2\eta \quad (17)$$

909

□

B Additional Information

Definition B.1. (Reversibility [17]) A neural network ϕ that maps the input \mathbf{x} to output $\mathbf{y} = \phi(\mathbf{x}; \theta)$ is reversible, if there exists $L(\mathbf{x}; \theta)$ so that $\mathbf{y} = L(\mathbf{x}; \theta)\mathbf{x}$, and the backpropagated gradient \mathbf{g}_x satisfies $\mathbf{g}_x = L^\top(\mathbf{x}; \theta)\mathbf{g}_y$, where \mathbf{g}_y is the backpropagated gradient at the output \mathbf{y} . $L(\mathbf{x}; \theta)$ depends on the input \mathbf{x} and weight θ in the network ϕ .

Several critical observations regarding Algorithm 1 warrant attention. Initially, in order to minimize memory consumption, Algorithm 1 implements a per-layer weight update during the process of backpropagation, as advocated by contemporary studies, see, e.g., [44]. This approach contrasts with conventional optimizers, which typically update all weights after backpropagation by retaining the complete gradients in memory, a method potentially marked by significant inefficiency. Should there be a desire to generate an adapter (i.e., a parallel low-dimensional LoRA-type model) subsequent to fine-tuning, this can be achieved with efficiency through the following steps. Firstly, the training weights gap $\Delta \triangleq \mathbf{W}_{\text{Fine-Tuned}} - \mathbf{W}_{\text{Pretrained}}$ is computed, where $\mathbf{W}_{\text{Fine-Tuned}}$ denotes the model weight upon process completion, and $\mathbf{W}_{\text{Pretrained}}$ refers to the original model weight. Subsequently, $r_{\text{Adaptor}} \triangleq \text{rank}(\Delta)$ is determined utilizing a matrix ranking algorithm, followed by the resolution of $\min_{\mathbf{A} \in \mathbb{R}^{n \times r_{\text{Adaptor}}}, \mathbf{B} \in \mathbb{R}^{r_{\text{Adaptor}} \times m}} \|\Delta - \mathbf{AB}\|_F^2$ through any optimization algorithm (e.g., gradient descent). It is noteworthy that any solution to this matrix factorization optimization problem is well-known as a global optimum [50].

C Update Step Rule Formulation

Definition C.1. [Subspace-Aware Moment-Orthogonalization (SUMO)] SUMO formulates the subsequent gradient update rules. Refer to

$$\text{SUMO} \begin{cases} \hat{\mathbf{G}}^{(t)} = \mathbf{Q}^{(t)\top} \nabla_{\mathbf{W}} f(\mathbf{W}_t; \xi_t) \mathbf{R}^{(t)} \\ \mathbf{M}^{(t+1)} = \beta \mathbf{M}^{(t)} + (1 - \beta) \hat{\mathbf{G}}^{(t)} \\ \mathbf{O}^{(t+1)} = \text{Orthogonalization_SVD}(\mathbf{M}^{(t+1)}) \\ \mathbf{W}^{(T)} = \mathbf{W}^{(0)} + \eta \sum_{t=0}^{T-1} \mathbf{Q}^{(t)} \mathbf{O}^{(t+1)} \mathbf{R}^{(t)\top}, \end{cases}$$

with $\mathbf{Q}_t \in \mathbb{R}^{m \times r}$ and $\mathbf{R}_t \in \mathbb{R}^{r \times n}$ denoting projection matrices, $T \in \mathbb{N}$ representing the subspace update interval, η indicating the learning rate, ξ_t constituting a stochastic batch, and $\text{Orthogonalization_SVD}(\mathbf{A})$ as the operator that resolves the following through Singular Value Decomposition (SVD), as described in

$$\arg \min_{\mathbf{O}} \{\|\mathbf{O} - \mathbf{A}\|_F : \text{either } \mathbf{O}^T \mathbf{O} = \mathbf{I} \text{ or } \mathbf{O} \mathbf{O}^T = \mathbf{I}\}.$$

D Additional Experiments

In Table 6, we evaluated SUMO and state-of-the-art memory-efficient fine-tuning methods on the MAWPS[51] dataset using the LLaMA2-7B model. We report results across two rank settings (32 and 128), comparing training time, memory usage, and task accuracy. SUMO consistently achieves superior accuracy while maintaining competitive efficiency in both memory and time (comparing to Galore).

Table 6: Fine-tuning LLaMA2-7B on MAWPS[51]

Methods	Rank	Time(h) ↓	Memory (GB) ↓	Accuracy (%) ↑
LoRA	32	0.40	14.36	45.80
DoRA	32	0.69	15.01	44.96
GaLore	32	2.59	15.15	58.40
SUMO (Newton-Shultz5)	32	1.83	13.86	58.47
SUMO (SVD)	32	1.56	13.86	61.23
LoRA	128	0.45	15.64	65.97
DoRA	128	0.72	16.17	66.81
GaLore	128	2.61	15.79	64.29
SUMO (Newton-Shultz5)	128	1.78	14.12	64.41
SUMO (SVD)	128	1.62	14.12	68.03

937 D.1 Details of Fine-Tuning on GLUE

938 We fine-tune the pre-trained RoBERTa-Base model on the GLUE benchmark using the model
 939 provided by the Hugging Face. In Table and, we detail the hyper parameters used in fine-tuning.

	MNLI	SST-2	MRPC	CoLA	QNLI	QQP	RTE	STS-B
Batch Size	16	16	16	32	16	16	16	16
# Epochs	30	30	30	30	30	30	30	30
Learning Rate	1E-05	1E-05	3E-05	3E-05	1E-05	1E-05	1E-05	1E-05
Rank Config.				$r = 4$				
Projection back scale				4				
Max Seq. Len.				512				

Table 7: Hyperparameters of fine-tuning RoBERTa base for the comparison in Table 2 with respect only to rank=4.

	MNLI	SST-2	MRPC	CoLA	QNLI	QQP	RTE	STS-B
Batch Size	16	16	16	32	16	16	16	16
# Epochs	30	30	30	30	30	30	30	30
Learning Rate	1E-05	2E-05	2E-05	1E-05	1E-05	2E-05	2E-05	3E-05
Rank Config.				$r = 8$				
Projection back scale				2				
Max Seq. Len.				512				

Table 8: Hyperparameters of fine-tuning RoBERTa base for the comparison in Table 2 with respect only to rank=8.