RefLoRA: Refactored Low-Rank Adaptation for Efficient Fine-Tuning of Large Models

Yilang Zhang
Department of ECE
University of Minnesota
Minneapolis, MN 55414
zhan7453@umn.edu

Bingcong Li
Department of CS
ETH Zürich
8092 Zürich, Switzerland
bingcong.li@inf.ethz.ch

Georgios B. Giannakis
Department of ECE
University of Minnesota
Minneapolis, MN 55414
georgios@umn.edu

Abstract

Low-Rank Adaptation (LoRA) lowers the computational and memory overhead of fine-tuning large models by updating a low-dimensional subspace of the pre-trained weight matrix. Albeit efficient, LoRA exhibits suboptimal convergence and noticeable performance degradation, due to inconsistent and imbalanced weight updates induced by its nonunique low-rank factorizations. To overcome these limitations, this article identifies the optimal low-rank factorization per step that minimizes an upper bound on the loss. The resultant refactored low-rank adaptation (RefLoRA) method promotes a flatter loss landscape, along with consistent and balanced weight updates, thus speeding up stable convergence. Extensive experiments evaluate RefLoRA on natural language understanding, and commonsense reasoning tasks with popular large language models including DeBERTaV3, LLaMA-7B, LLaMA2-7B and LLaMA3-8B. The numerical tests corroborate that RefLoRA converges faster, outperforms various benchmarks, and enjoys negligible computational overhead compared to state-of-the-art LoRA variants.

1 Introduction

Large language models (LLMs) have revolutionized a wide spectrum of applications including chatbots [1], code generation [7], and scientific discovery [54]. Despite their success, adapting LLMs to specific tasks remains computationally demanding. LLMs are built on a two-stage learning process, namely *pre-training* and *fine-tuning*. Pre-training is performed on massive, Internet-scale corpora. This endows LLMs with in-text comprehension and generation abilities, but results in models with billions to trillions parameters [1, 16]. Though broad language capabilities are granted, pre-trained LLMs are yet not tailored for specialized applications. To acquire domain-specific expertise, LLMs must be further trained on downstream tasks, what is known as fine-tuning. With the continually growing model size however, conventional full fine-tuning approaches (optimizing all model parameters) can be increasingly prohibitive due to the immense GPU memory and substantial computational capacity demands, rendering them impossible for individual users and organizations.

To tackle the computational bottleneck, parameter-efficient fine-tuning (PEFT) [21] has been investigated to enhance fine-tuning efficiency. As opposed to fully fine-tuning all parameters, PEFT methods either optimize a sparse subset [17, 53], or introduce additional lightweight trainable parameters while keeping the pre-trained ones frozen [21, 45, 35, 28, 32]. Among these approaches, low-rank adaptation (LoRA) [22] has gained popularity due to its low additional cost during inference. LoRA presumes that the parameter updates during fine-tuning lie on a low-dimensional manifold, and thus can be captured by a low-rank matrix. Though effective, LoRA is observed to suffer from challenges such as slow and unstable convergence [40], inconsistent and unbalanced weight updates [66, 19], and

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

notable performance gaps relative to full fine-tuning [22]. One key reason behind these challenges is the nonuniqueness of LoRA's low-rank factorization [66].

To cope with these challenges, this work advocates "refactoring" low-rank adaption (RefLoRA), a novel approach that commits to the optimal factorization per step. Our contribution is threefold:

- We show that LoRA's inconsistent weight updates can be characterized by a symmetric positive definite matrix. RefLoRA dynamically selects the optimal one by minimizing an upper bound on the loss, resulting in flatter landscape of the loss that facilitates stable and efficient optimization.
- The optimal factorization is proven to have a closed-form global solution; to yield consistent and balanced weight updates; and to bring about a lower overhead compared to SOTA approaches. Moreover, a simplified variant termed RefLoRA-S is developed to further reduce complexity.
- Extensive numerical tests are conducted on matrix factorization, natural language understanding, and commonsense reasoning benchmarks with popular LLMs scaled up to 8B parameters, demonstrating RefLoRA's faster convergence and consistent performance gain.

Related work. Building upon LoRA, several have been developed to ameliorate its performance. Some revise vanilla LoRA's architecture or dynamically adjust its hyperparameters. For instance, DoRA [64] decomposes LoRA weights into magnitude and direction components to improve learning capacity and training stability. AdaLoRA [69] and GeoLoRA [50] allocates per-layer rank on-the-fly to prune less salient updates. FedPara [24] (a.k.a. LoHa) and LoKr [65] respectively integrate the low-rank structure with Hadamard and Kronecker products for reduced communication cost and improved model expressiveness. Another line of work boosts LoRA's behavior in early fine-tuning epochs through well-designed initialization. PiSSA [40] leverages truncated singular value decomposition (SVD) to keep the low-rank initialization close to the pre-trained weights, whereas LoRA-GA [59] aligns the first update direction with full fine-tuning. Other variants refine LoRA's per-step optimization to promote empirical convergence. LoRA-Pro [60] redirects LoRA's gradient to match the weight update of full fine-tuning. LoRA-RITE [66] substitutes the default Adam optimizer [25] with customized gradient calculation and moment estimation scheme. However, these gradient-altering strategies necessitate meticulous crafting to ensure stability and convergence. Our approach falls within the same family, but adheres strictly to standard backpropagation rule, avoiding direct gradient manipulation and requiring less computational overhead. To meet constraints of space limits, additional LoRA variants are discussed in Appendix A.

2 Preliminaries and nonunique low-rank factorizations

Consider a general weight matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$ parameterizing a large model. With initialization being the pre-trained matrix $\mathbf{W}_0 := \mathbf{W}^{\rm pt}$ and t indexing iteration, conventional full fine-tuning updates the sizable matrix $\mathbf{W}_t$ by backpropagating the loss function $\ell(\mathbf{W}_t)$. Though straightforward, this approach requires an excessive memory footprint and major computational cost.

Aiming at efficient fine-tuning, LoRA [22] freezes $\mathbf{W}^{\rm pt}$ and presumes that the weight increment exhibits a low-rank structure $\mathbf{W}_t = \mathbf{W}^{\rm pt} + \mathbf{A}_t\mathbf{B}_t^\top$, where $\mathbf{A}_t \in \mathbb{R}^{m\times r}$, $\mathbf{B}_t \in \mathbb{R}^{n\times r}$, and $r \ll \min\{m, n\}$ is a preselected constant. Consequently, LoRA's objective function boils down to

$$\min_{\mathbf{A},\mathbf{B}} \mathcal{L}(\mathbf{A}, \mathbf{B}) := \ell(\mathbf{W}^{\rm pt} + \mathbf{A}\mathbf{B}^\top).$$

This reformulation reduces the number of trainable parameters to $(m + n)r \ll mn$, thus effectively minimizing the associated memory and computation costs. As for initialization, LoRA draws entries of $\mathbf{A}_0$ from a zero-mean normal distribution with small variance σ^2 for numerical stability; and $\mathbf{B}_0 = \mathbf{0}$ to ensure $\mathbf{W}_0 = \mathbf{W}^{\rm pt}$. This choice gives rise to imbalanced updates of $\mathbf{A}_t$ and $\mathbf{B}_t$, and decelerates empirical convergence especially in early epochs. At the first iteration for example, the chain rule implies that $\nabla_{\mathbf{A}_0}\mathcal{L}(\mathbf{A}_0, \mathbf{B}_0) = \nabla\ell(\mathbf{W}^{\rm pt} + \mathbf{A}_0\mathbf{B}_0^\top)\mathbf{B}_0 = \mathbf{0}$, meaning no update to $\mathbf{A}_0$. In comparison, $\mathbf{B}_0$'s gradient $\nabla_{\mathbf{B}_0}\mathcal{L}(\mathbf{A}_0, \mathbf{B}_0) = \nabla\ell(\mathbf{W}^{\rm pt})^\top\mathbf{A}_0$ is generally non-zero. In fact, it turns out that $\|\mathbf{B}_t\|_{\rm F} \ll \|\mathbf{A}_t\|_{\rm F}$ and $\|\nabla_{\mathbf{A}_t}\mathcal{L}(\mathbf{A}_t, \mathbf{B}_t)\|_{\rm F} \ll \|\nabla_{\mathbf{B}_t}\mathcal{L}(\mathbf{A}_t, \mathbf{B}_t)\|_{\rm F}$ when t is small [40], and these unbalanced low-rank factors and updates can markedly impede the convergence of LoRA.

Additionally, the *nonuniqueness* of LoRA's low-rank factors leads to inconsistent parameter updates. While this issue has been previously recognized [42, 66], its implications for optimization remain underexplored. Specifically, for any alternative decomposition $(\tilde{\mathbf{A}}_t, \tilde{\mathbf{B}}_t)$ satisfying $\tilde{\mathbf{A}}_t\tilde{\mathbf{B}}_t^\top = \mathbf{A}_t\mathbf{B}_t^\top$,

the forward pass and loss remain intact, whereas the update of $\mathbf{W}_t$ can differ significantly. For illustration, consider standard gradient descent (GD) iterations

$$\mathbf{A}_{t+1} = \mathbf{A}_t + \Delta\mathbf{A}_t := \mathbf{A}_t - \eta\nabla\ell(\mathbf{W}_t)\mathbf{B}_t, \quad \mathbf{B}_{t+1} = \mathbf{B}_t + \Delta\mathbf{B}_t := \mathbf{B}_t - \eta\nabla\ell(\mathbf{W}_t)^\top\mathbf{A}_t \quad (1)$$

where $\eta > 0$ denotes the learning rate. The corresponding update to the weight matrix is

$$\Delta\mathbf{W}_t := \mathbf{W}_{t+1} - \mathbf{W}_t = \mathbf{A}_{t+1}\mathbf{B}_{t+1}^\top - \mathbf{A}_t\mathbf{B}_t^\top = \mathbf{A}_t\Delta\mathbf{B}_t^\top + \Delta\mathbf{B}_t\mathbf{A}_t^\top + \Delta\mathbf{A}_t\Delta\mathbf{B}_t^\top. \quad (2)$$

Alternatively, GD can be also performed with the equivalent pair $(\tilde{\mathbf{A}}_t, \tilde{\mathbf{B}}_t)$, yielding

$$\mathbf{A}_{t+1} = \tilde{\mathbf{A}}_t + \Delta\tilde{\mathbf{A}}_t := \tilde{\mathbf{A}}_t - \eta\nabla\ell(\mathbf{W}_t)\tilde{\mathbf{B}}_t, \quad \mathbf{B}_{t+1} = \tilde{\mathbf{B}}_t + \Delta\tilde{\mathbf{B}}_t := \tilde{\mathbf{B}}_t - \eta\nabla\ell(\mathbf{W}_t)^\top\tilde{\mathbf{A}}_t. \quad (3)$$

The resultant parameter update is thereby $\Delta\tilde{\mathbf{W}}_t := \tilde{\mathbf{A}}_t\Delta\tilde{\mathbf{B}}_t^\top + \Delta\tilde{\mathbf{B}}_t\tilde{\mathbf{A}}_t^\top + \Delta\tilde{\mathbf{A}}_t\Delta\tilde{\mathbf{B}}_t^\top$, which can remarkably deviate from $\Delta\mathbf{W}_t$ in (2), despite both factorizations representing the same $\mathbf{W}_t$. Further elaboration on this issue will be provided in the ensuing section.

The following notational conventions are adopted throughout the paper.

Notation. Bold lowercase (capital) letters denote vectors (matrices); $\|\cdot\|_2$, $\|\cdot\|_*$, $\|\cdot\|_{\mathrm{F}}$ and $\langle\cdot,\cdot\rangle_{\mathrm{F}}$ stand for ℓ_2-, nuclear-, Frobenius-norm and Frobenius inner product; $\mathrm{Col}(\cdot)$, $\mathrm{Null}(\cdot)$ and $[\cdot]_i$ represent column space, null space and the i-th entry/column of a vector/matrix; $\cdot^\dagger$, $\mathrm{tr}(\cdot)$, and $\mathrm{rank}(\cdot)$ refer to the Moore-Penrose pseudoinverse, trace, and rank; $\lambda_i(\cdot)$ and $\sigma_i(\cdot)$ are the i-th largest eigenvalue and singular value; $\mathbb{S}_{++}^r$ indicates the set of $r \times r$ symmetric positive definite (SPD) matrices; $\mathrm{GL}(r)$ denotes the general linear group of degree r (i.e., the set of $r \times r$ invertible matrices); and $\mathrm{O}(r)$ stands for the orthogonal group of degree r (i.e., the set of $r \times r$ orthogonal matrices).

3 Low-rank adaptation with optimal refactoring

This section delves into the nonuniqueness of LoRA's factorization, and identifies the optimal one that minimizes the loss $\ell(\mathbf{W}_t + \Delta\tilde{\mathbf{W}}_t)$. It is first demonstrated that *all* possible factors $(\tilde{\mathbf{A}}_t, \tilde{\mathbf{B}}_t)$ can be characterized by an invertible matrix $\mathbf{P}_t$, and the weight update $\Delta\tilde{\mathbf{W}}_t$ is fundamentally governed by an SPD matrix $\mathbf{S}_t := \mathbf{P}_t\mathbf{P}_t^\top$. Then, we will derive an upper bound of $\ell(\mathbf{W}_t + \Delta\tilde{\mathbf{W}}_t)$ as a function of $\mathbf{S}_t$, whose global minimum will be obtained in closed form. Building on these theoretical insights, we will optimally refactor LoRA's low-rank matrices per iteration to obtain more effective updates, that are at the center of our "refactored" low-rank adaptation (RefLoRA) approach. All proofs in this section are deferred to Appendix B.

3.1 Characterizing LoRA's factorization and weight update

Our analysis begins with the following mild assumption, which has been utilized and validated on various realistic datasets [60].

Assumption 1. $\mathrm{rank}(\mathbf{A}_t) = \mathrm{rank}(\mathbf{B}_t) = r$, $\forall t > 0$.

Assumption 1 asserts that the tall matrices $\mathbf{A}_t$ and $\mathbf{B}_t$ maintain full column rank after the first iteration. Since LoRA seeks to approximate the full update of $\mathbf{W}_t$ using a low-rank matrix, this assumption essentially reflects the effectiveness of LoRA's parameterization. Under Assumption 1, the next lemma reveals that *all* equivalent factorizations can be captured by an $r \times r$ invertible matrix $\mathbf{P}_t$.

Lemma 1. *With Assumption 1 in effect, it holds that*

$$\{(\tilde{\mathbf{A}}_t, \tilde{\mathbf{B}}_t) \mid \tilde{\mathbf{A}}_t\tilde{\mathbf{B}}_t^\top = \mathbf{A}_t\mathbf{B}_t^\top\} = \{(\mathbf{A}_t\mathbf{P}_t, \mathbf{B}_t\mathbf{P}_t^{-\top}) \mid \mathbf{P}_t \in \mathrm{GL}(r)\}. \quad (4)$$

Moreover, if $\mathbf{P}_t \in \mathrm{O}(r)$*, then* $\Delta\tilde{\mathbf{W}}_t = \Delta\mathbf{W}_t$.

Beyond characterizing the structure of equivalent factorizations, Lemma 1 implies that consistent weight updates are preserved when $(\tilde{\mathbf{A}}_t, \tilde{\mathbf{B}}_t)$ differs from $(\mathbf{A}_t, \mathbf{B}_t)$ up to a rotation and reflection. This naturally raises the question: *which factorization, or* $\mathbf{P}_t$*, is preferable for effective optimization?*

To answer the last question, we first present an important observation about $\Delta\tilde{\mathbf{W}}_t$. When setting $\tilde{\mathbf{A}}_t = \mathbf{A}_t\mathbf{P}_t$ and $\tilde{\mathbf{B}}_t = \mathbf{B}_t\mathbf{P}_t^{-\top}$, it can be readily verified (see (12) of Appendix B.1) that

$$\Delta\tilde{\mathbf{W}}_t = \mathbf{A}_t(\mathbf{P}_t\mathbf{P}_t^\top)\Delta\mathbf{B}_t^\top + \Delta\mathbf{A}_t(\mathbf{P}_t\mathbf{P}_t^\top)^{-1}\mathbf{B}_t^\top + \Delta\mathbf{A}_t\Delta\mathbf{B}_t^\top. \quad (5)$$

Letting $\mathbf{P}_t = \mathbf{U}_t^P \mathbf{\Sigma}_t^P \mathbf{V}_t^{P\top}$ denote the SVD of $\mathbf{P}_t$, it is clear that $\Delta\tilde{\mathbf{W}}_t$ is fully determined by the $r \times r$ SPD matrix $\mathbf{S}_t := \mathbf{P}_t\mathbf{P}_t^\top = \mathbf{U}_t^P(\mathbf{\Sigma}_t^P)^2\mathbf{U}_t^{P\top} \in \mathbb{S}^r_{++}$. This implies that $\mathbf{V}_t^P$ can be chosen arbitrarily from $\mathrm{O}(r)$. In other words, $\mathbf{P}_t$ can be right-multiplied by any orthogonal matrix, without affecting $\Delta\tilde{\mathbf{W}}_t$. This observation agrees with the last statement of Lemma 1 by replacing $(\mathbf{A}_t, \mathbf{B}_t)$ and $\mathbf{P}_t$ with $(\mathbf{A}_t\mathbf{U}_t^P\mathbf{\Sigma}_t^P, \mathbf{B}_t\mathbf{U}_t^P(\mathbf{\Sigma}_t^P)^{-1})$ and $\mathbf{V}_t^P$. Consequently, identifying the optimal factorization boils down to selecting an ideal $\mathbf{S}_t \in \mathbb{S}^r_{++}$.

3.2 Optimizing $\mathbf{S}_t$ via loss upper bound minimization

Our key idea is to select the $\mathbf{S}_t$ that minimizes the loss $\ell(\mathbf{W}_{t+1}) = \ell(\mathbf{W}_t + \Delta\tilde{\mathbf{W}}_t(\mathbf{S}_t))$. Unfortunately, directly minimizing this objective over $\mathbf{S}_t$ requires exhaustive search due to the model nonlinearity, which is infeasible in practice. As an alternative, we derive a tractable upper bound on $\ell(\mathbf{W}_t + \Delta\tilde{\mathbf{W}}_t(\mathbf{S}_t))$, and minimize the bound to obtain an optimal $\mathbf{S}_t$. Our motivation stems from GD, which relies on the following Lipschitz smoothness assumption.

Assumption 2. *The loss function ℓ has L-Lipschitz gradient; i.e., $\|\nabla\ell(\mathbf{W}) - \nabla\ell(\mathbf{W}')\|_{\mathrm{F}} \leq L\|\mathbf{W} - \mathbf{W}'\|_{\mathrm{F}},\ \forall \mathbf{W}, \mathbf{W}' \in \mathbb{R}^{m\times n}$.*

Assumption 2 is equivalent to requiring Lipschitz smoothness of ℓ w.r.t. the vectorized weight matrix $\mathrm{vec}(\mathbf{W})$, which is fairly mild and common in machine learning [51, 15] and optimization [3, 5]. Under this assumption, the loss admits the following quadratic upper bound

$$\ell(\mathbf{W}_t + \Delta\mathbf{W}_t) \leq \ell(\mathbf{W}_t) + \langle\nabla\ell(\mathbf{W}_t), \Delta\mathbf{W}_t\rangle_{\mathrm{F}} + \frac{L}{2}\|\Delta\mathbf{W}_t\|_{\mathrm{F}}^2. \tag{6}$$

Minimizing the bound yields the optimum $\Delta\mathbf{W}_t^* = -\frac{1}{L}\nabla\ell(\mathbf{W}_t)$, thus recovering the standard GD used in full fine-tuning. Given that L is typically unknown in practice, the optimal learning rate $1/L$ is replaced by a hyperparameter, whose value can be tuned via grid search on a validation dataset.

Although ℓ is often Lipschitz smooth w.r.t. $\mathbf{W}_t$, its smoothness constants w.r.t. $\mathbf{A}_t$ and $\mathbf{B}_t$ can be unbounded due to the bilinear structure, unless one assumes boundedness or convergence of $\mathbf{A}_t$ and $\mathbf{B}_t$ [13]. Since these two assumptions are overly restrictive, they will be avoided in our analysis.

The nonlinear dependence of $\Delta\tilde{\mathbf{W}}_t$ on $\mathbf{S}_t$ (cf. (5)) prevents an analytical solution when directly optimizing $\mathbf{S}_t$ over the quadratic upper bound of $\ell(\mathbf{W}_t + \Delta\tilde{\mathbf{W}}_t(\mathbf{S}_t))$. This motivates iterative solvers involving matrix multiplication with the sizable matrix $\nabla\ell(\mathbf{W}_t) \in \mathbb{R}^{m\times n}$. To mitigate the overhead, the next proposition relaxes the quadratic upper bound to factor out $\|\nabla\ell(\mathbf{W}_t)\|_2^2$, thus decoupling $\mathbf{S}_t$'s optimization from $\nabla\ell(\mathbf{W}_t)$.

Proposition 2. *Consider GD update* (3) *with $\tilde{\mathbf{A}}_t = \mathbf{A}_t\mathbf{P}_t$, $\tilde{\mathbf{B}}_t = \mathbf{B}_t\mathbf{P}_t^{-\top}$, and $\mathbf{S}_t := \mathbf{P}_t\mathbf{P}_t^\top$. Under Assumptions 1 and 2, it follows that*

$$\ell(\mathbf{W}_t + \Delta\tilde{\mathbf{W}}_t(\mathbf{S}_t)) \leq \frac{L\eta^2}{2}\|\nabla\ell(\mathbf{W}_t)\|_2^2\Big(\|\mathbf{A}_t\mathbf{S}_t^{\frac{1}{2}}\|_{\mathrm{F}}^2 + \|\mathbf{B}_t\mathbf{S}_t^{-\frac{1}{2}}\|_{\mathrm{F}}^2 - \frac{1}{L\eta}\Big)^2 + \mathcal{O}(L\eta^3) + \mathrm{Const.} \tag{7}$$

where Const. *refers to constants that do not rely on $\mathbf{S}_t$.*

The high-order term $\mathcal{O}(L\eta^3)$ originates from $\Delta\mathbf{A}_t\Delta\mathbf{B}_t^\top$ in (5). As η is typically small ($\sim \mathcal{O}(10^{-4})$), this term is negligible in practice [59, 66]. As a consequence, the upper bound in (7) is dominated by its first term. This leads to our RefLoRA objective

$$\min_{\mathbf{S}_t \in \mathbb{S}^r_{++}} \Big(\|\mathbf{A}_t\mathbf{S}_t^{\frac{1}{2}}\|_{\mathrm{F}}^2 + \|\mathbf{B}_t\mathbf{S}_t^{-\frac{1}{2}}\|_{\mathrm{F}}^2 - \frac{1}{L\eta}\Big)^2. \tag{8}$$

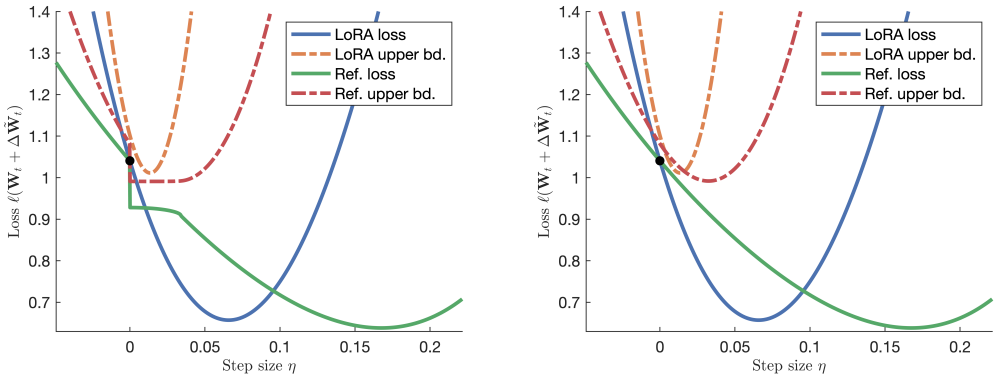
Consider for convenience the variables $\tilde{\mathbf{S}}_t$ and $\tilde{C}_t$ defined as

$$\tilde{\mathbf{S}}_t := (\mathbf{A}_t^\top\mathbf{A}_t)^{-\frac{1}{2}}\big[(\mathbf{A}_t^\top\mathbf{A}_t)^{\frac{1}{2}}\mathbf{B}_t^\top\mathbf{B}_t(\mathbf{A}_t^\top\mathbf{A}_t)^{\frac{1}{2}}\big]^{\frac{1}{2}}(\mathbf{A}_t^\top\mathbf{A}_t)^{-\frac{1}{2}}, \quad \tilde{C}_t := 2\|\mathbf{A}_t\mathbf{B}_t^\top\|_* \tag{9}$$

based on which (8) can be solved in closed form as established in the following theorem.

Theorem 3. *Under Assumptions 1 and 2, the global optimum of* (8) *satisfies*

$$\mathbf{S}_t^* \begin{cases} = \tilde{\mathbf{S}}_t, & \text{if } \eta \geq \frac{1}{\tilde{C}_t L} \text{ or } \eta < 0 \\ \ni \left[(\tilde{C}_t L\eta)^{-1} \pm \sqrt{(\tilde{C}_t L\eta)^{-2} - 1}\right]\tilde{\mathbf{S}}_t, & \text{if } 0 < \eta < \frac{1}{\tilde{C}_t L} \end{cases}. \tag{10}$$



(a) $\mathbf{S}_t = \mathbf{S}_t^*$ (b) $\mathbf{S}_t = \tilde{\mathbf{S}}_t$

Figure 1: Visualization of loss $\ell(\mathbf{W}_t + \Delta\tilde{\mathbf{W}}_t)$ and upper bound (7). LoRA corresponds to $\mathbf{S}_t = \mathbf{I}_r$, while our refactoring (ref.) optimizes $\mathbf{S}_t$.

Theorem 3 states that if $\eta > 0$ is not too small, then (8) admits a unique global optimum $\tilde{\mathbf{S}}_t$. Otherwise, there can be multiple optima, while two can always be constructed by appropriately scaling $\tilde{\mathbf{S}}_t$. In addition, $\eta < 0$ is included in (10) for visualization purposes. We remark that $\tilde{\mathbf{S}}_t$ in (9) is known as the matrix geometric mean $(\mathbf{A}_t^\top \mathbf{A}_t)^{-1} \# (\mathbf{B}_t^\top \mathbf{B}_t)$ of $(\mathbf{A}_t^\top \mathbf{A}_t)^{-1}$ and $(\mathbf{B}_t^\top \mathbf{B}_t)$ [31], which can be also written as $\tilde{\mathbf{S}}_t = (\mathbf{A}_t^\top \mathbf{A}_t)^{-1} [\mathbf{A}_t^\top \mathbf{A}_t \mathbf{B}_t^\top \mathbf{B}_t]^{\frac{1}{2}}$; cf. Lemma 7 in the Appendix.

Figure 1 plots the loss $\ell(\mathbf{W}_t + \Delta\tilde{\mathbf{W}}_t)$, and the upper bound (7) of a numerical example as a function of η; see also Appendix D.2 for details. LoRA corresponds to the non-optimized $\mathbf{S}_t = \mathbf{I}_r$, whereas our refactoring selects $\mathbf{S}_t = \mathbf{S}_t^*$ based on Theorem 3. Notably, $\eta = 0$ leads to a jump discontinuity for our refactoring. Figure 1a shows that by optimizing the upper bound (7), the associated loss becomes lower and flatter. This enables a larger step size η to achieve a lower loss, thereby accelerating the empirical convergence of LoRA; see Section 4 for experiments corroborating this claim.

The flatter loss landscape arises thanks to balancing $\tilde{\mathbf{A}}_t$ and $\tilde{\mathbf{B}}_t$. Indeed, the imbalance of $\mathbf{A}_t$ and $\mathbf{B}_t$ in vanilla LoRA necessitates different step sizes [19]. Appendix B.3 proves that the balance $\tilde{\mathbf{A}}_t^\top \tilde{\mathbf{A}}_t = \tilde{\mathbf{B}}_t^\top \tilde{\mathbf{B}}_t$ is guaranteed with $\mathbf{S}_t = \tilde{\mathbf{S}}_t$, thus enabling a unified step size. With η too small however, it is more beneficial (compared to balanced updates) to scale either $\mathbf{A}_t$ or $\mathbf{B}_t$ to accommodate the small η. This corresponds to the two solutions in the $0 < \eta < 1/(\tilde{C}_t L)$ case of (10).

3.3 RefLoRA: Refactored low-rank adaptation

Having identified in Theorem 3 the optimal $\mathbf{S}_t^*$ minimizing the loss upper bound, we are ready to introduce our refactored low-rank adaptation (RefLoRA) approach. RefLoRA substitutes LoRA's per-step update (1) with the refactored version (3), where $\tilde{\mathbf{A}}_t = \mathbf{A}_t \mathbf{P}_t$, $\tilde{\mathbf{B}}_t = \mathbf{B}_t \mathbf{P}_t^{-\top}$, and $\mathbf{P}_t \mathbf{P}_t^\top = \mathbf{S}_t^*$. As the right singular matrix $\mathbf{V}_t^P \in \mathrm{O}(r)$ can be arbitrary (cf. Section 3.1), one convenient choice is to simply set $\mathbf{P}_t = \mathbf{S}_t^{*1/2}$. Next, this subsection deals with two practical challenges facing the implementation of RefLoRA, and delves into several important properties of the resultant approach.

Smoothness constant L is typically unknown and difficult to estimate in practice, especially for LLMs. Thus, it is unclear when to switch between the two schemes in (10). As a remedy, one can either treat $1/L$ as a hyperparameter akin to GD, or, adhere to the balanced update by choosing $\mathbf{S}_t = \tilde{\mathbf{S}}_t$ for $\forall \eta$. The latter results in a continuous loss function and upper bounds as sketched in Figure 1b. Since this adjustment only affects the region where η is tiny, it still allows for a larger η to improve convergence. For simplicity, the balanced update is adopted thereafter.

Adaptive optimizers such as Adam [25] and AdamW [39] are default to optimizing large models, which adjust the update using the first moment and entrywise second moment estimated from the running average of stochastic gradients. When refactoring $(\mathbf{A}_t, \mathbf{B}_t)$ to $(\mathbf{A}_t \mathbf{P}_t, \mathbf{B}_t \mathbf{P}_t^{-\top})$, the first moment estimator can be transformed accordingly, while the second moment is generally intractable due to the entrywise square. To tackle this challenge, we "refactor back" the low-rank matrices after

Table 1: Additional complexities introduced by LoRA variants

Method	Time	Space
LoRA forward/backward	$\Omega(mn)$	$\Omega(mn)$
LoRA-Pro [60]	$\mathcal{O}(m^2r + (m+n+r)r^2)$	$\mathcal{O}(m^2 + (m+n+r)r)$
LoRA-RITE [66]	$\mathcal{O}((m+n+r)r^2)$	$\mathcal{O}((m+n+r)r)$
RefLoRA (Thm. 3)	$\mathcal{O}((m+n+r)r^2)$	$\mathcal{O}(r^2)$
RefLoRA-S (Thm. 5)	$\mathcal{O}((m+n)r)$	$\mathcal{O}(1)$

the GD update in (3) by right-multiplying $\mathbf{P}_t^{-1}$ and $\mathbf{P}_t^\top$. This gives an alternative update

$$\mathbf{A}_{t+1}{:=}(\tilde{\mathbf{A}}_t{+}\Delta\tilde{\mathbf{A}}_t)\mathbf{P}_t^{-1}{=}\mathbf{A}_t{-}\eta\nabla\ell(\mathbf{W}_t)\mathbf{B}_t\tilde{\mathbf{S}}_t^{-1},\ \mathbf{B}_{t+1}{:=}(\tilde{\mathbf{B}}_t{+}\Delta\tilde{\mathbf{B}}_t)\mathbf{P}_t^\top{=}\mathbf{B}_t{-}\eta\nabla\ell(\mathbf{W}_t)^\top\mathbf{A}_t\tilde{\mathbf{S}}_t$$

whose axes align with the original $(\mathbf{A}_t, \mathbf{B}_t)$, thus waiving the need to transform the moment estimators. This observation prompts the view of refactoring as GD with preconditioning matrices $\tilde{\mathbf{S}}_t^{-1}$ and $\tilde{\mathbf{S}}_t$. As a side benefit, this also eliminates the need to compute $\mathbf{P}_t$ from $\tilde{\mathbf{S}}_t$.

Next, we present three key RefLoRA properties.

Balanced refactoring $\mathbf{A}_t^\top\mathbf{A}_t = \mathbf{B}_t^\top\mathbf{B}_t$ can be achieved per iteration upon setting $\mathbf{P}_t\mathbf{P}_t^\top = \tilde{\mathbf{S}}_t$, as stated in Section 3.2. It is worthwhile pointing out that SVD-based initializations including PiSSA [40] and LoftQ [33] inherently satisfy $\mathbf{A}_0^\top\mathbf{A}_0 = \mathbf{B}_0^\top\mathbf{B}_0^\top$. When t is small, the balance tends to hold approximately even without refactoring, which partially explains why empirical convergence is fast during the early epochs. Analytically, balanced refactoring maximizes the potential loss reduction as formalized in the ensuing Theorem.

Theorem 4. *The solution* $\mathbf{S}_t = \tilde{\mathbf{S}}_t$ *given by* (9) *minimizes the lower bound*

$$0 \geq \underbrace{\left\langle \nabla_{\tilde{\mathbf{A}}_t}\ell(\tilde{\mathbf{W}}_t), \Delta\tilde{\mathbf{A}}_t\right\rangle_{\mathrm{F}} + \left\langle \nabla_{\tilde{\mathbf{B}}_t}\ell(\tilde{\mathbf{W}}_t), \Delta\tilde{\mathbf{B}}_t\right\rangle_{\mathrm{F}}}_{\approx\Delta\ell(\mathbf{W}_t):=\ell(\mathbf{W}_{t+1})-\ell(\mathbf{W}_t)\ \textit{with small}\ \eta} \geq -\eta\|\nabla\ell(\mathbf{W}_t)\|_2^2\big(\|\mathbf{A}_t\mathbf{S}_t^{\frac{1}{2}}\|_{\mathrm{F}}^2 + \|\mathbf{B}_t\mathbf{S}_t^{-\frac{1}{2}}\|_{\mathrm{F}}^2\big).$$

The upper bound 0 stems from the low-rank nature of LoRA, which can be reached when $\nabla\ell(\mathbf{W}_t)^\top \in \mathrm{Null}(\mathbf{B}_t^\top)$ and $\nabla\ell(\mathbf{W}_t) \in \mathrm{Null}(\mathbf{A}_t^\top)$; i.e., the stationary points of $\mathbf{A}_t$ and $\mathbf{B}_t$. While this upper bound cannot be improved, RefLoRA minimizes the lower bound, and thus leads to a more effective descent in the loss.

Additional computational overhead induced by RefLoRA is as small as $\mathcal{O}((m+n+r)r^2)$ in time and $\mathcal{O}(r^2)$ in memory, thanks to the decoupling of $\nabla\ell(\mathbf{W}_t)$ in Proposition (2). Compared to the forward/backward overhead $\Omega(mn)$ of LoRA, the extra complexity introduced by RefLoRA is relatively minimal. In contrast, LoRA-Pro [60] suffers from $\mathcal{O}(m^2r + (m+n+r)r^2)$ time and $\mathcal{O}(m^2 + (m+n+r)r)$ space, whereas LoRA-RITE [66] requires $\mathcal{O}((m+n+r)r)$ memory for polar decomposition. Despite the low complexity of RefLoRA, further curtailing the extra cost can be beneficial for resource-limited applications. The next theorem restricts $\mathbf{S}_t$ to a scaled identity $s_t\mathbf{I}_r$ with $s_t > 0$, and derives a result analogous to Theorem 3.

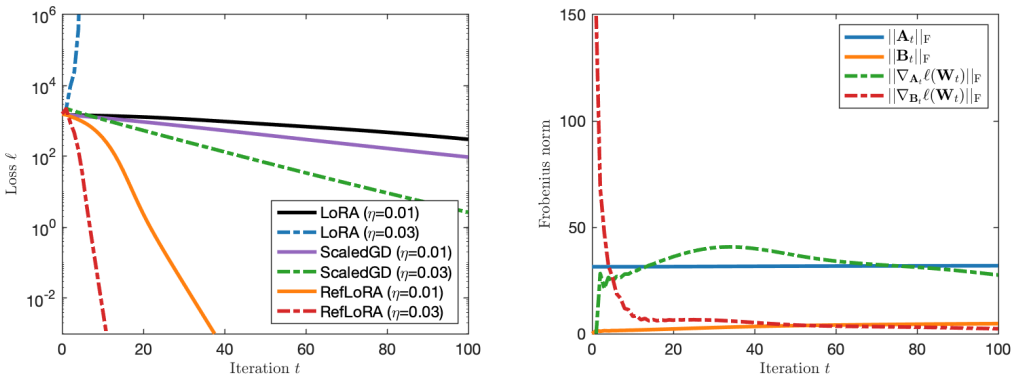
Theorem 5. *Consider* (8) *confined with* $\mathbf{S}_t = s_t\mathbf{I}_r,\ s_t \in \mathbb{R}_{++}$. *Under Assumptions 1 and 2, it holds*

$$s_t^* = \begin{cases} \frac{\|\mathbf{B}_t\|_{\mathrm{F}}}{\|\mathbf{A}_t\|_{\mathrm{F}}}, & \textit{if}\ \eta \geq \frac{1}{2\|\mathbf{A}_t\|_{\mathrm{F}}\|\mathbf{B}_t\|_{\mathrm{F}}L}\ \textit{or}\ \eta < 0 \\ \frac{\frac{1}{L\eta} \pm \sqrt{\frac{1}{L^2\eta^2} - 4\|\mathbf{A}_t\|_{\mathrm{F}}^2\|\mathbf{B}_t\|_{\mathrm{F}}^2}}{2\|\mathbf{A}_t\|_{\mathrm{F}}^2}, & \textit{if}\ 0 < \eta < \frac{1}{2\|\mathbf{A}_t\|_{\mathrm{F}}\|\mathbf{B}_t\|_{\mathrm{F}}L} \end{cases}. \tag{11}$$

This simplified refactoring (RefLoRA-S) enjoys further reduced complexity of $\mathcal{O}((m+n)r)$ time and $\mathcal{O}(1)$ space; see Table 1 for a summary. Compared to Theorem 3, the scalar s_t only has two global optima when η is too small. Note that this simplification only guarantees a weaker balance $\|\tilde{\mathbf{A}}_t\|_{\mathrm{F}} = \|\tilde{\mathbf{B}}_t\|_{\mathrm{F}}$. However, it can be directly combined with adaptive optimizers by scaling the moment estimators without the second refactoring.

Consistent weight updates are always guaranteed for all equivalent factorizations.

Theorem 6. *Under Assumptions 1-2 and for any* $\mathbf{A}_t'\mathbf{B}_t'^\top = \mathbf{A}_t\mathbf{B}_t^\top$, *let* $\Delta\tilde{\mathbf{W}}_t'$ *and* $\Delta\tilde{\mathbf{W}}_t$ *be the corresponding weight updates* (5) *with RefLoRA. It then always holds that* $\Delta\tilde{\mathbf{W}}_t' = \Delta\tilde{\mathbf{W}}_t$.



(a) Loss ℓ vs. iteration t (b) Norms of LoRA's factors and gradients

Figure 2: Comparison of LoRA, ScaledGD, and RefLoRA for matrix factorization

This consistency holds for either $\mathbf{S}_t = \mathbf{S}_t^*$ or $\mathbf{S}_t = \tilde{\mathbf{S}}_t$, but not for the lightweight version (11). In this context, we dealt with slow convergence and the non-uniqueness of low-rank factorization with the added benefit of balanced updates. The step-by-step algorithm of RefLoRA(-S) is listed in Appendix C. Next, experiments are conducted to verify our findings.

4 Numerical tests

This section evaluates the empirical performance of RefLoRA. All experimental setups including platforms, datasets, models, metrics, and hyperparameters are detailed in Appendix D. Our implementation is available at `https://github.com/zhangyilang/RefLoRA`.

4.1 Matrix factorization

The first numerical test considers low-rank matrix factorization [13]

$$\min_{\mathbf{A},\mathbf{B}} \frac{1}{2}\|\mathbf{Y} - \mathbf{A}\mathbf{B}\|_{\mathrm{F}}^2$$

where $\mathbf{Y} \in \mathbb{R}^{m\times n}$ is a given low-rank matrix. It can be viewed as applying LoRA to a single-layer model, and training it with whitened data [2, 30]. Figure 2a compares the training loss of RefLoRA with LoRA, and another popular approach dubbed ScaledGD [55]—which is tailored particularly for low-rank matrix factorization, and has gained popularity among LoRA variants; see e.g., [68]. Each method is tested with two learning rates $\eta \in \{0.01, 0.03\}$. While vanilla LoRA converges slowly with the lower learning rate and diverges with the higher one, RefLoRA remains stable and converges markedly faster under both rates, as corroborated also with observations in Figure 1. Notably, RefLoRA even outperforms ScaledGD, thanks to its balanced update. Figure 2b depicts the dynamics of LoRA ($\eta = 0.01$) by plotting the Frobenius norms of $\mathbf{A}_t, \mathbf{B}_t$ and their gradients. It is observed that $\|\mathbf{A}_t\|_{\mathrm{F}}$ and $\|\mathbf{B}_t\|_{\mathrm{F}}$ are highly unbalanced across iterations, and $\|\nabla_{\mathbf{A}_t}\ell\|_{\mathrm{F}}$ as well as $\|\nabla_{\mathbf{B}_t}\ell\|_{\mathrm{F}}$ exhibit a sharp change when t is small. In comparison, RefLoRA maintains $\tilde{\mathbf{A}}_t^\top \tilde{\mathbf{A}}_t = \tilde{\mathbf{B}}_t^\top \tilde{\mathbf{B}}_t$, $\forall t$, which guarantees that $\|\tilde{\mathbf{A}}_t\|_{\mathrm{F}}^2 = \mathrm{tr}(\tilde{\mathbf{A}}_t^\top \tilde{\mathbf{A}}_t) = \mathrm{tr}(\tilde{\mathbf{B}}_t^\top \tilde{\mathbf{B}}_t) = \|\tilde{\mathbf{B}}_t\|_{\mathrm{F}}$. This balance can afford larger learning rates, thus improving the empirical convergence.

4.2 Natural language understanding

Beyond matrix factorization the evaluation here starts with fine-tuning DeBERTaV3-base [20], a masked language model with 184M parameters, on the General Language Understanding Evaluation (GLUE) benchmark [58]. GLUE contains 8 datasets, providing a general-purpose evaluation for natural language understanding (NLU) [58]. The test setup follows from [22, 69], where LoRA rank is $r = 8$, reducing the trainable parameters to 1.33M. We compare RefLoRA and its lightweight variant RefLoRA-S against a suite of LoRA variants, including SOTA methods DoRA [64] and AdaLoRA [69], as well as baselines falling in the same category with RefLoRA, i.e., LoRA-Pro [60]

Table 2: Performance comparison using DeBERTaV3-base on the GLUE benchmark dataset. The best results are depicted in solid lines. The score in the last column averages Matthews correlation coefficient (Mcc), accuracies (Acc), Pearson correlation (Corr), and matched accuracy (M). The results are obtained by averaging 5 random runs.

Method	**Params**	**CoLA**	**SST-2**	**MRPC**	**STS-B**	**QQP**	**MNLI**	**QNLI**	**RTE**	**All**
		Mcc	Acc	Acc	Corr	Acc/F1	M/Mm	Acc	Acc	Avg
Full FT	184M	69.19	95.63	89.46	91.60	92.40/89.80	89.90/90.12	94.03	83.75	88.25
BitFit	0.1M	66.96	94.84	87.75	91.35	88.41/84.95	89.37/89.91	92.24	78.70	86.20
HAdapter	1.22M	68.64	95.53	89.95	91.48	91.91/89.27	90.13/90.17	94.11	84.48	88.28
PAdapter	1.18M	68.77	95.61	89.46	91.54	92.04/89.40	90.33/90.39	94.29	85.20	88.41
LoRA	1.33M	69.82	94.95	89.95	91.60	91.99/89.38	90.65/90.69	93.87	85.20	88.50
DoRA	1.33M	70.85	95.79	90.93	91.79	92.07/-	90.29/-	94.10	86.04	88.98
AdaLoRA	1.27M	71.45	**96.10**	90.69	91.84	92.23/89.74	**90.76/90.79**	**94.55**	88.09	89.46
LoRA-Pro	1.33M	71.36	95.76	90.20	91.92	92.19/89.60	90.23/90.19	94.29	85.56	88.94
LoRA-RITE	1.33M	69.55	95.41	90.93	91.79	92.02/89.42	90.22/90.33	94.42	85.20	88.69
RefLoRA	1.33M	**71.73**	95.99	**91.42**	92.03	92.28/89.70	90.23/90.41	94.40	**88.09**	**89.52**
RefLoRA-S	1.33M	70.66	95.76	90.44	**92.21**	**92.43/89.89**	90.13/90.17	94.16	87.73	89.19

and LoRA-RITE [66]. The results can be found in Table 2. It is worth mentioning that these datasets are relatively small, so that full fine-tuning (FT) is prone to overfitting, thus leading to worse performance compared to PEFT methods. RefLoRA and RefLoRA-S outperform all competitors on 5 out of 8 datasets, and present comparable performance to SOTA approaches on the rest 3 datasets. Overall, RefLoRA achieves the highest average performance, demonstrating more effective optimization via refactoring. In spite of the simplified refactoring, RefLoRA-S maintains competitive performance, i.e., only 0.33% lower than RefLoRA on average, while markedly reducing computational overhead.

4.3 Commonsense reasoning

We further extend our numerical experiments to fine-tuning the LLaMA series [56, 57, 16], which are autoregressive language models with 7B and 8B parameters. We tackle commonsense reasoning tasks following the setup in [23, 64]. Training data are aggregated from 8 datasets listed in Table 3, and test sets remain separate for individual evaluation. These reasoning tasks are intended to push the model beyond pattern recognition, requiring commonsense and knowledge to make proper inferences. The baselines are chosen as DoRA [64], LoRA-RITE [66], PrecLoRA [68], and NoRA+ [30]. Note that the latter two approaches are variants of ScaledGD [55], sharing similar complexity with RefLoRA. The accuracy comparison is summarized in Table 3. Both RefLoRA and RefLoRA-S consistently outperform other PEFT methods in 5 out of 6 settings. Even under lower-rank configurations $r = 16$, both RefLoRA and RefLoRA-S continue to lead or match top-performing approaches, underscoring their parameter efficiency and robustness. These results demonstrate the effectiveness of the proposed RefLoRA(-S), and underscore the potential of optimal refactoring.

4.4 Subject-driven image generation with diffusion models

Akin to LoRA, RefLoRA can be seamlessly integrated into a wide range of larger models beyond LLMs. Further numerical tests are conducted on a subject-driven image generation task [47] with Stable Diffusion v1.4 [44]. The goal is to fine-tune a diffusion model using a few user-provided images so that it can generate the same object in various contexts. Specifically, the model is fine-tuned on a set of images labeled "a photo of sks dog," and subsequently evaluated by generating images under the prompt "a sks dog eating nachos." LoRA adapters with rank $r = 4$ are attached to the U-Net component of Stable Diffusion, and experimental setups including hyperparameters are default to those in [47]. The fine-tuning losses for LoRA, LoRA-Pro, LoRA-RITE, and RefLoRA are summarized in the table 4, where average and standard deviation are calculated over 3 random runs. It is seen that RefLoRA achieves 14.0% , 13.1%, and 9.5% improvements over LoRA, LoRA-Pro, and LoRA-RITE, respectively. In addition to quantitative gains in loss, we also observe noticeable improvements in image quality; see Figure 3. The generations from RefLoRA-tuned models show markedly clearer details and better object fidelity, particularly in the mouth and tongue regions, where features are often distorted in outputs from other three baselines.

Table 3: Accuracy comparison using LLaMA series on commonsense reasoning datasets.

	r	Method	Params	BoolQ	PIQA	SIQA	HS	WG	ARCe	ARCc	OBQA	Avg
ChatGPT-3.5-turbo			-	73.1	85.4	68.5	78.5	66.1	89.8	79.9	74.8	77.0
LLaMA-7B	32	LoRA	0.83%	66.42	80.03	77.84	82.88	81.85	79.92	63.40	77.20	76.19
		PrecLoRA	0.83%	68.96	80.95	77.43	81.54	80.27	78.83	64.16	79.20	76.42
		NoRA+	0.83%	69.85	81.83	77.38	82.09	80.03	79.67	64.25	78.60	76.71
		DoRA	0.84%	69.7	83.4	78.6	87.2	81.0	81.9	66.2	79.2	78.4
		LoRA-RITE	0.84%	69.82	82.75	78.55	84.72	81.69	82.15	66.23	81.40	78.54
		RefLoRA	0.83%	69.60	82.48	79.53	88.25	82.56	81.57	66.64	80.20	**78.85**
		RefLoRA-S	0.83%	70.18	82.48	78.15	87.41	82.08	81.52	65.36	81.60	78.60
	16	DoRA	0.43%	70.0	82.6	79.7	83.2	80.6	80.6	65.4	77.6	77.5
		RefLoRA	0.41%	69.66	82.43	79.43	87.38	81.22	80.68	65.44	78.60	**78.11**
		RefLoRA-S	0.41%	67.65	81.50	79.07	88.28	81.77	81.23	64.59	78.60	77.84
LLaMA2-7B	32	LoRA	0.83%	69.8	79.9	79.5	83.6	82.6	79.8	64.7	81.0	77.6
		PrecLoRA	0.83%	71.47	81.50	78.81	85.97	80.43	81.14	66.55	81.00	78.36
		NoRA+	0.83%	70.52	81.94	79.07	87.66	82.24	82.70	67.06	80.20	78.92
		DoRA	0.84%	71.8	83.7	76.0	89.1	82.6	83.7	68.2	82.4	79.7
		LoRA-RITE	0.84%	71.04	82.43	79.79	89.12	84.53	83.88	68.77	81.20	80.10
		RefLoRA	0.83%	72.54	83.79	80.04	86.94	84.85	86.36	71.50	80.20	**80.78**
		RefLoRA-S	0.83%	73.36	83.84	80.76	90.02	82.48	84.55	67.92	82.60	80.69
	16	DoRA	0.43%	72.0	83.1	79.9	89.1	83.0	84.5	71.0	81.2	**80.5**
		RefLoRA	0.41%	71.38	82.43	80.35	90.49	83.43	84.05	69.28	82.00	80.43
		RefLoRA-S	0.41%	72.08	83.03	80.45	85.89	83.27	84.30	69.88	82.00	80.11
LLaMA3-8B	32	LoRA	0.70%	70.8	85.2	79.9	91.7	84.3	84.2	71.2	79.0	80.8
		PrecLoRA	0.70%	70.73	85.80	78.86	91.87	83.66	85.10	71.08	82.40	81.19
		NoRA+	0.70%	71.16	85.10	79.48	92.22	83.35	85.86	72.27	83.20	81.58
		DoRA	0.71%	74.6	89.3	79.9	95.5	85.6	90.5	80.4	85.8	85.2
		LoRA-RITE	0.84%	74.19	89.44	81.52	95.44	86.74	90.45	80.12	86.60	85.56
		RefLoRA	0.70%	75.35	88.74	80.91	95.71	86.66	90.49	80.20	87.40	85.68
		RefLoRA-S	0.70%	75.50	89.72	81.11	95.59	87.29	90.99	79.78	86.00	**85.75**
	16	DoRA	0.35%	74.5	88.8	80.3	95.5	84.7	90.1	79.1	87.2	85.0
		RefLoRA	0.35%	75.26	88.79	81.37	95.85	85.64	90.11	80.55	86.60	**85.52**
		RefLoRA-S	0.35%	74.92	89.01	80.60	95.75	85.24	90.45	80.89	86.40	85.41

Table 4: Fine-tuning loss (↓) for subject-driven image generation on DreamBooth.

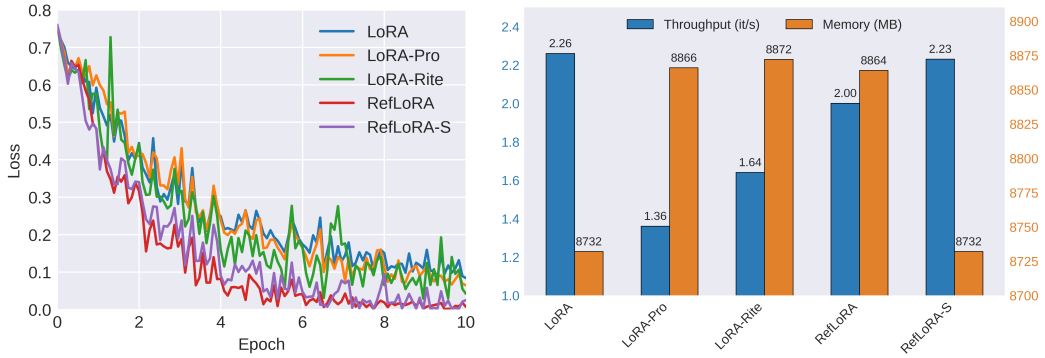
Loss	LoRA	LoRA-Pro	LoRA-RITE	RefLoRA
Avg±std	0.100 ± 0.015	0.099 ± 0.015	0.095 ± 0.016	$\mathbf{0.086} \pm 0.017$

4.5 Convergence and complexity comparison

Lastly, we evaluate the convergence behavior and computational efficiency of RefLoRA in comparison with LoRA [22], LoRA-Pro [60], and LoRA-RITE [66]. These tests are conducted on the MRPC subset of the GLUE benchmark using DeBERTaV3-base as the backbone model. For fairness, the learning rate is set to $\eta = 4 \times 10^{-4}$ across all methods. Figure 4a depicts the loss $\ell(\mathbf{W}_t)$ over 10 fine-tuning epochs. The loss of RefLoRA(-S) declines more rapidly and exhibits less fluctuation than the other three methods, which ultimately achieves the lowest value approaching 0. This confirms stability and convergence speed, on par with our theoretical insights in Section 3. The sharper descent in the loss suggests improved optimization trajectories enabled by RefLoRA's principled refactoring.

The comparison of computational overhead is presented in Figure 4b. Time complexity is reflected via the fine-tuning throughput (iterations per second; higher is better), while space complexity is measured in GPU memory occupation (lower is better). For better visualization, the vertical axes start with a non-zero value. The plot reveals that RefLoRA and RefLoRA-S respectively showcase 88.5% and 98.7% throughput compared to LoRA, at the additional memory cost of 132MB and < 1MB. In contrast, the throughput of LoRA-Pro and LoRA-RITE are 60.2% and 72.6% of LoRA, requiring 134MB and 140MB extra memory. This is consistent with our complexity analysis in Table 1. Extended comparisons scaling up to 27B models are offered in Appendix E.

Figure 3: Images generated from Stable Diffusion fine-tuned with different approaches.



(a) Loss ℓ vs. epoch

(b) Throughput (↑) and GPU memory usage (↓)

Figure 4: Convergence and complexity comparison

5 Conclusion and outlook

This paper introduced refactored low-rank adaptation (RefLoRA), a principled LoRA variant that remarkably enhances the efficiency and stability of fine-tuning large models. By identifying the optimal matrix $\mathbf{S}_t$ that minimizes the loss upper bound, RefLoRA addressed the key challenges in LoRA by providing balanced updates, improved convergence, and consistently superior empirical performance with affordable overhead. To further facilitate scalability and applicability to large models, RefLoRA-S leverages simplified refactoring to minimize the computational complexity. Our future research agenda involves analyzing the convergence rate of LoRA and RefLoRA, and adapting RefLoRA to extensive model architectures such as vision transformers and diffusion models.

Acknowledgments

This work is partly supported by NSF grants 2220292, 2212318, 2312547, and 2332547. B. Li is supported by Swiss National Science Foundation (SNSF) Project Funding No. 200021-207343.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2018.

[3] Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997.

[4] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proc. AAAI Conf. Artif. Intel.*, pages 7432–7439, 2020.

[5] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[6] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. In *Proc. Int. Workshop Semant. Eval.*, pages 1–14. ACL, 2017.

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

[8] François Chollet. On the measure of intelligence. *arXiv:1911.01547*, 2019.

[9] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, June 2019.

[10] René Descartes. *La Géométrie*. Jan Maire, Leiden, 1637.

[11] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 36, pages 10088–10115, 2023.

[12] Bill Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proc. Int. Workshop Paraphrasing*, 2005.

[13] Simon S Du, Wei Hu, and Jason D Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

[14] Ziqi Gao, Qichao Wang, Aochuan Chen, Zijing Liu, Bingzhe Wu, Liang Chen, and Jia Li. Parameter-efficient fine-tuning with discrete fourier transform. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2024.

[15] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[16] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[17] Demi Guo, Alexander M Rush, and Yoon Kim. Parameter-efficient transfer learning with diff pruning. *arXiv preprint arXiv:2012.07463*, 2020.

[18] Yongchang Hao, Yanshuai Cao, and Lili Mou. FLORA: Low-rank adapters are secretly gradient compressors. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2024.

[19] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low rank adaptation of large models. In *Proc. Int. Conf. on Machine Learning (ICML)*, volume 235, pages 17783–17806, 21–27 Jul 2024.

[20] Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2023.

[21] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proc. Int. Conf. on Machine Learning (ICML)*, volume 97, pages 2790–2799, 09–15 Jun 2019.

[22] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2022.

[23] Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. LLM-Adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[24] Nam Hyeon-Woo, Moon Ye-Bin, and Tae-Hyun Oh. FedPara: Low-rank hadamard product for communication-efficient federated learning. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2022.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2015.

[26] Toshiaki Koike-Akino, Francesco Tonin, Yongtao Wu, Frank Zhengqing Wu, Leyla Naz Candogan, and Volkan Cevher. Quantum-peft: Ultra parameter-efficient fine-tuning. *arXiv preprint arXiv:2503.05431*, 2025.

[27] Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M Asano. VeRA: Vector-based random matrix adaptation. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2024.

[28] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3045–3059, November 2021.

[29] Bingcong Li, Liang Zhang, and Niao He. Implicit regularization of sharpness-aware minimization for scale-invariant problems. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 37, pages 44444–44478, 2024.

[30] Bingcong Li, Liang Zhang, Aryan Mokhtari, and Niao He. On the crucial role of initialization for matrix factorization. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2025.

[31] Chi-Kwong Li and Roy Mathias. Geometric means. *Linear algebra and its applications*, 385:305–334, 2004.

[32] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proc. Conf. Assoc. Comput. Linguist. Meet. (ACL)*, pages 4582–4597, August 2021.

[33] Yixiao Li, Yifan Yu, Chen Liang, Nikos Karampatziakis, Pengcheng He, Weizhu Chen, and Tuo Zhao. LoftQ: LoRA-fine-tuning-aware quantization for large language models. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2024.

[34] Vladislav Lialin, Sherin Muckatira, Namrata Shivagunde, and Anna Rumshisky. ReLoRA: High-rank training through low-rank updates. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2024.

[35] Zhaojiang Lin, Andrea Madotto, and Pascale Fung. Exploring versatile generative language model via parameter-efficient transfer learning. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 441–459, November 2020.

[36] Vijay Lingam, Atula Tejaswi, Aditya Vavre, Aneesh Shetty, Gautham Krishna Gudur, Joydeep Ghosh, Alex Dimakis, Eunsol Choi, Aleksandar Bojchevski, and Sujay Sanghavi. Svft: Parameter-efficient fine-tuning with singular vectors. *arXiv:2405.19597*, 2024.

[37] Weiyang Liu, Zeju Qiu, Yao Feng, Yuliang Xiu, Yuxuan Xue, Longhui Yu, Haiwen Feng, Zhen Liu, Juyeon Heo, Songyou Peng, et al. Parameter-efficient orthogonal finetuning via butterfly factorization. *arXiv preprint arXiv:2311.06243*, 2023.

[38] Ziyue Liu, Ruijie Zhang, Zhengyang Wang, Zi Yang, Paul Hovland, Bogdan Nicolae, Franck Cappello, and Zheng Zhang. Cola: Compute-efficient pre-training of llms via low-rank activation. *arXiv preprint arXiv:2502.10940*, 2025.

[39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2019.

[40] Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 37, pages 121038–121072, 2024.

[41] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. *arXiv:1809.02789*, 2018.

[42] Theo Putterman, Derek Lim, Yoav Gelberg, Stefanie Jegelka, and Haggai Maron. Learning on loras: Gl-equivariant processing of low-rank weight spaces for large finetuned models. *arXiv:2410.04207*, 2024.

[43] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proc. Conf. Assoc. Comput. Linguist. Meet. (ACL)*, pages 784–789, 2018.

[44] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[45] Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. AdapterDrop: On the efficiency of adapters in transformers. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7930–7946, November 2021.

[46] Walter Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, New York, 3rd edition, 1976.

[47] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 22500–22510, June 2023.

[48] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

[49] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv:1904.09728*, 2019.

[50] Steffen Schotthöfer, Emanuele Zangrando, Gianluca Ceruti, Francesco Tudisco, and Jonas Kusch. GeoloRA: Geometric integration for parameter efficient fine-tuning. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2025.

[51] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[52] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642, 2013.

[53] Yi-Lin Sung, Varun Nair, and Colin A Raffel. Training neural networks with fixed sparse masks. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 34, pages 24193–24205, 2021.

[54] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.

[55] Tian Tong, Cong Ma, and Yuejie Chi. Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent. *J. Mach. Learn. Res.*, 22(150):1–63, 2021.

[56] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

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

[58] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2019.

[59] Shaowen Wang, Linxi Yu, and Jian Li. Lora-ga: Low-rank adaptation with gradient approximation. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 37, pages 54905–54931, 2024.

[60] Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. LoRA-pro: Are low-rank adapters properly optimized? In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2025.

[61] Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *Trans. Assoc. Comput. Linguist.*, 7:625–641, 2019.

[62] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proc. Conf. North Am. Chapter Assoc. Comput. Linguist.*, pages 1112–1122, 2018.

[63] Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Reft: Representation finetuning for language models. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 37, pages 63908–63962, 2024.

[64] Shih yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2024.

[65] SHIH-YING YEH, Yu-Guan Hsieh, Zhidong Gao, Bernard B W Yang, Giyeong Oh, and Yanmin Gong. Navigating text-to-image customization: From LyCORIS fine-tuning to model evaluation. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2024.

[66] Jui-Nan Yen, Si Si, Zhao Meng, Felix Yu, Sai Surya Duvvuri, Inderjit S Dhillon, Cho-Jui Hsieh, and Sanjiv Kumar. LoRA done RITE: Robust invariant transformation equilibration for loRA optimization. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2025.

[67] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv:1905.07830*, 2019.

[68] Fangzhao Zhang and Mert Pilanci. Riemannian preconditioned LoRA for fine-tuning foundation models. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2024.

[69] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2023.

A Additional related work

LoRA variants. LoRA has been extended in several directions. For example, recent works further reduce LoRA's trainable parameters [27, 36, 14, 18, 26]. As these methods can achieve performance comparable to that of LoRA, they indirectly suggest that the expressiveness of LoRA has not been fully exploited. Another line of work [11, 33] incorporates quantization in LoRA to reduce memory footprint and computational overhead. There are also works that broaden the applicability of LoRA to pre-training LLMs by e.g., sequentially chaining LoRA modules [34, 38]. As our method enhances the efficiency of LoRA training, we expect that it can be seamlessly integrated to such settings as well. Going beyond LoRA, there are also approaches for fine-tuning LLMs in a parameter efficient manner such as [32, 28, 37, 63]. Moreover, [29] shows that applying sharpness-aware minimization on LoRA promotes balance between $\mathbf{A}_t$ and $\mathbf{B}_t$. These approaches are orthogonal to our work.

Broader Impact. The theoretical insights and algorithmic contributions of the current work are broadly applicable across a range of fine-tuning scenarios. Our RefLoRA enhances the efficiency and effectiveness of adapting language models to downstream tasks, leading to improved performance in applications such as sentiment classification. This, in turn, can positively impact real-world systems including recommendation systems by increasing accuracy and relevance. However, caution should be exercised when deploying the method for generative tasks. In such settings, the outputs of language models should be carefully reviewed, and proper safeguards, such as gating mechanisms, should be considered to ensure safety, reliability, and trustworthiness of the generated content.

Future directions. Due to limited computational resources, our evaluation currently deals with models having reasonably large scale, e.g., LLaMA3-8B. Our future work will include scaling RefLoRA to even larger models, such as those with 30B parameters. Another promising direction is to integrate RefLoRA with sequentially chaining, namely [34]. This will further broaden the applicability of RefLoRA for pre-training LLMs.

B Missing proofs

This appendix presents the proofs that were omitted from the main paper.

B.1 Proof of Lemma 1

Proof. First, we prove (4) by showing that the two sets in (4) contain each other.

For any pair $(\mathbf{A}_t\mathbf{P}_t, \mathbf{B}_t\mathbf{P}_t^{-\top})$, it is easy to see $\mathbf{A}_t\mathbf{P}_t(\mathbf{B}_t\mathbf{P}_t^{-\top})^\top = \mathbf{A}_t\mathbf{B}_t^\top$. Thus we have

$$\{(\tilde{\mathbf{A}}_t, \tilde{\mathbf{B}}_t) \mid \tilde{\mathbf{A}}_t\tilde{\mathbf{B}}_t^\top = \mathbf{A}_t\mathbf{B}_t^\top\} \supseteq \{(\mathbf{A}_t\mathbf{P}_t, \mathbf{B}_t\mathbf{P}_t^{-\top}) \mid \mathbf{P}_t \in \mathrm{GL}(r)\}.$$

Next, we prove the opposite containing relationship. Let $(\tilde{\mathbf{A}}_t, \tilde{\mathbf{B}}_t)$ be an arbitrary pair satisfying $\tilde{\mathbf{A}}_t\tilde{\mathbf{B}}_t^\top = \mathbf{A}_t\mathbf{B}_t^\top$. It follows that

$$\begin{aligned}
&\mathrm{rank}(\tilde{\mathbf{A}}_t\tilde{\mathbf{B}}_t^\top) = \mathrm{rank}(\mathbf{A}_t\mathbf{B}_t^\top) \geq \mathrm{rank}(\mathbf{A}_t) + \mathrm{rank}(\mathbf{B}_t) - r = r,\\
&\mathrm{rank}(\tilde{\mathbf{A}}_t\tilde{\mathbf{B}}_t^\top) \leq \min\{\mathrm{rank}(\tilde{\mathbf{A}}_t), \mathrm{rank}(\tilde{\mathbf{B}}_t)\} \leq r.
\end{aligned}$$

We thus obtain $\mathrm{rank}(\mathbf{A}_t\mathbf{B}_t^\top) = \mathrm{rank}(\tilde{\mathbf{A}}_t\tilde{\mathbf{B}}_t^\top) = r$, and $\mathrm{rank}(\tilde{\mathbf{A}}_t) = \mathrm{rank}(\tilde{\mathbf{B}}_t) = r$.

Since $\mathrm{Col}(\mathbf{A}_t\mathbf{B}_t^\top) \subseteq \mathrm{Col}(\mathbf{A}_t)$ and $\dim(\mathrm{Col}(\mathbf{A}_t\mathbf{B}_t^\top)) = \mathrm{rank}(\mathbf{A}_t\mathbf{B}_t^\top) = r = \mathrm{rank}(\mathbf{A}_t) = \dim(\mathrm{Col}(\mathbf{A}_t))$, we have $\mathrm{Col}(\mathbf{A}_t\mathbf{B}_t^\top) = \mathrm{Col}(\mathbf{A}_t)$; and likewise $\mathrm{Col}(\tilde{\mathbf{A}}_t\tilde{\mathbf{B}}_t^\top) = \mathrm{Col}(\tilde{\mathbf{A}}_t)$.

Then, the condition $\tilde{\mathbf{A}}_t\tilde{\mathbf{B}}_t^\top = \mathbf{A}_t\mathbf{B}_t^\top$ leads to

$$\mathrm{Col}(\tilde{\mathbf{A}}_t) = \mathrm{Col}(\tilde{\mathbf{A}}_t\tilde{\mathbf{B}}_t^\top) = \mathrm{Col}(\mathbf{A}_t\mathbf{B}_t^\top) = \mathrm{Col}(\mathbf{A}_t).$$

This suggests, there must exist an invertible matrix $\mathbf{P}_t \in \mathbb{R}^{r\times r}$ such that $\tilde{\mathbf{A}}_t = \mathbf{A}_t\mathbf{P}_t$.

As a result, we have $\mathbf{A}_t\mathbf{P}_t\tilde{\mathbf{B}}_t^\top = \mathbf{A}_t\mathbf{B}_t^\top$. Multiplying $\mathbf{P}_t^{-1}\mathbf{A}_t^\dagger$ on both sides and taking transpose yield $\tilde{\mathbf{B}}_t = \mathbf{B}_t\mathbf{P}_t^{-\top}$. This suggests

$$\{(\tilde{\mathbf{A}}_t, \tilde{\mathbf{B}}_t) \mid \tilde{\mathbf{A}}_t\tilde{\mathbf{B}}_t^\top = \mathbf{A}_t\mathbf{B}_t^\top\} \subseteq \{(\mathbf{A}_t\mathbf{P}_t, \mathbf{B}_t\mathbf{P}_t^{-\top}) \mid \mathbf{P}_t \in \mathrm{GL}(r)\}.$$

which completes the proof of (4).

Additionally, it follows from the definition of $\Delta\tilde{\mathbf{W}}_t$ that

$$\begin{aligned}
\Delta\tilde{\mathbf{W}}_t &= \tilde{\mathbf{A}}_t\Delta\tilde{\mathbf{B}}_t^\top + \Delta\tilde{\mathbf{A}}_t\tilde{\mathbf{B}}_t^\top + \Delta\tilde{\mathbf{A}}_t\Delta\tilde{\mathbf{B}}_t^\top \\
&\overset{(a)}{=} -\eta\tilde{\mathbf{A}}_t\tilde{\mathbf{A}}_t^\top\nabla\ell(\mathbf{W}_t) - \eta\nabla\ell(\mathbf{W}_t)\tilde{\mathbf{B}}_t\tilde{\mathbf{B}}_t^\top + \eta^2\nabla\ell(\mathbf{W}_t)\tilde{\mathbf{B}}_t\tilde{\mathbf{A}}_t^\top\nabla\ell(\mathbf{W}_t) \\
&\overset{(b)}{=} -\eta\mathbf{A}_t\mathbf{P}_t\mathbf{P}_t^\top\mathbf{A}_t^\top\nabla\ell(\mathbf{W}_t) - \eta\nabla\ell(\mathbf{W}_t)\mathbf{B}_t\mathbf{P}_t^{-\top}\mathbf{P}_t^{-1}\mathbf{B}_t^\top + \eta^2\nabla\ell(\mathbf{W}_t)\mathbf{B}_t\mathbf{A}_t^\top\nabla\ell(\mathbf{W}_t) \\
&\overset{(c)}{=} \mathbf{A}_t(\mathbf{P}_t\mathbf{P}_t^\top)\Delta\mathbf{B}_t^\top + \Delta\mathbf{A}_t(\mathbf{P}_t\mathbf{P}_t^\top)^{-1}\mathbf{B}_t^\top + \Delta\mathbf{A}_t\Delta\mathbf{B}_t^\top
\end{aligned} \tag{12}$$

where (a) uses (3) and that $\mathbf{W}_t^{\mathrm{pt}} + \tilde{\mathbf{A}}_t\tilde{\mathbf{B}}_t^\top = \mathbf{W}_t^{\mathrm{pt}} + \mathbf{A}_t\mathbf{B}_t^\top = \mathbf{W}_t$, (b) is due to $\tilde{\mathbf{A}}_t = \mathbf{A}_t\mathbf{P}_t$ and $\tilde{\mathbf{B}}_t = \mathbf{B}_t\mathbf{P}^{-\top}$, and (c) leverages (1).

Comparing (12) with (2), it can be easily seen that $\mathbf{P}_t\mathbf{P}_t^\top = \mathbf{I}_r$ results in $\Delta\mathbf{W}_t = \Delta\tilde{\mathbf{W}}_t$. □

B.2 Proof of Proposition 2

Proof. For (3), it follows from Assumption 2 that

$$\begin{aligned}
&\ell(\mathbf{W}_t + \Delta\tilde{\mathbf{W}}_t(\mathbf{S}_t)) \le \ell(\mathbf{W}_t) + \langle\nabla\ell(\mathbf{W}_t), \Delta\tilde{\mathbf{W}}_t(\mathbf{S}_t)\rangle_{\mathrm{F}} + \frac{L}{2}\|\Delta\tilde{\mathbf{W}}_t(\mathbf{S}_t)\|_{\mathrm{F}}^2 \\
&\overset{(a)}{=} \ell(\mathbf{W}_t) + \big\langle\nabla\ell(\mathbf{W}_t), \mathbf{A}_t\mathbf{S}_t\Delta\mathbf{B}_t^\top + \Delta\mathbf{A}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top - \Delta\mathbf{A}_t\Delta\mathbf{B}_t^\top\big\rangle_{\mathrm{F}} + \\
&\quad \frac{L}{2}\big\|\mathbf{A}_t\mathbf{S}_t\Delta\mathbf{B}_t^\top + \Delta\mathbf{A}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top - \Delta\mathbf{A}_t\Delta\mathbf{B}_t^\top\big\|_{\mathrm{F}}^2 \\
&\overset{(b)}{=} \big\langle\nabla\ell(\mathbf{W}_t), \mathbf{A}_t\mathbf{S}_t\Delta\mathbf{B}_t^\top + \Delta\mathbf{A}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\big\rangle_{\mathrm{F}} + \frac{L}{2}\Big[\big\|\mathbf{A}_t\mathbf{S}_t\Delta\mathbf{B}_t^\top\big\|_{\mathrm{F}}^2 + \big\|\Delta\mathbf{A}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\big\|_{\mathrm{F}}^2 + \\
&\quad 2\big\langle\mathbf{A}_t\mathbf{S}_t\Delta\mathbf{B}_t^\top, \Delta\mathbf{A}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\big\rangle_{\mathrm{F}} - 2\big\langle\mathbf{A}_t\mathbf{S}_t\Delta\mathbf{B}_t^\top + \Delta\mathbf{A}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top, \Delta\mathbf{A}_t\Delta\mathbf{B}_t^\top\big\rangle_{\mathrm{F}}\Big] + \mathrm{Const.} \\
&\overset{(c)}{=} \frac{L}{2}\Big[\Big\|\frac{1}{L}\nabla\ell(\mathbf{W}_t) + \mathbf{A}_t\mathbf{S}_t\Delta\mathbf{B}_t^\top\Big\|_{\mathrm{F}}^2 + \Big\|\frac{1}{L}\nabla\ell(\mathbf{W}_t) + \Delta\mathbf{A}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\Big\|_{\mathrm{F}}^2 + \\
&\quad 2\big\langle\mathbf{A}_t\mathbf{S}_t\Delta\mathbf{B}_t^\top, \Delta\mathbf{A}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\big\rangle_{\mathrm{F}} - 2\big\langle\mathbf{A}_t\mathbf{S}_t\Delta\mathbf{B}_t^\top + \Delta\mathbf{A}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top, \Delta\mathbf{A}_t\Delta\mathbf{B}_t^\top\big\rangle_{\mathrm{F}}\Big] + \mathrm{Const.}
\end{aligned} \tag{13}$$

where (a) relies on (12); (b) expands the squared Frobenius norm and merges terms independent $\mathbf{S}_t$ of into $\mathrm{Const.}$; and (c) utilizes completing the square and merges the constant terms.

Next, we bound the four terms in (13). Using the definition (1) of $\Delta\mathbf{B}_t$, the first term is relaxed via

$$\begin{aligned}
\Big\|\frac{1}{L}\nabla\ell(\mathbf{W}_t) + \mathbf{A}_t\mathbf{S}_t\Delta\mathbf{B}_t^\top\Big\|_{\mathrm{F}}^2 &= \Big\|\frac{1}{L}\nabla\ell(\mathbf{W}_t) - \eta\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top\nabla\ell(\mathbf{W}_t)\Big\|_{\mathrm{F}}^2 \\
&\le \eta^2\|\nabla\ell(\mathbf{W}_t)\|_2^2\Big\|\frac{1}{L\eta}\mathbf{I}_m - \mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top\Big\|_{\mathrm{F}}^2.
\end{aligned} \tag{14}$$

Likewise, the second term is bounded through

$$\Big\|\frac{1}{L}\nabla\ell(\mathbf{W}_t) + \Delta\mathbf{A}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\Big\|_{\mathrm{F}}^2 \le \eta^2\|\nabla\ell(\mathbf{W}_t)\|_2^2\Big\|\frac{1}{L\eta}\mathbf{I}_n - \mathbf{B}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\Big\|_{\mathrm{F}}^2. \tag{15}$$

Again using the definitions of $\Delta\mathbf{A}_t$ and $\Delta\mathbf{B}_t$, the third term in (13) satisfies

$$\begin{aligned}
\big\langle\mathbf{A}_t\mathbf{S}_t\Delta\mathbf{B}_t^\top, \Delta\mathbf{A}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\big\rangle_{\mathrm{F}} &= \eta^2\big\langle\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top\nabla\ell(\mathbf{W}_t), \nabla\ell(\mathbf{W}_t)\mathbf{B}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\big\rangle_{\mathrm{F}} \\
&\overset{(a)}{\le} \eta^2\big\|\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top\nabla\ell(\mathbf{W}_t)\big\|_{\mathrm{F}}\big\|\nabla\ell(\mathbf{W}_t)\mathbf{B}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\big\|_{\mathrm{F}} \\
&\le \eta^2\|\nabla\ell(\mathbf{W}_t)\|_2^2\big\|\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top\big\|_{\mathrm{F}}\big\|\mathbf{B}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\big\|_{\mathrm{F}}
\end{aligned} \tag{16}$$

where (a) follows from Cauchy-Schwarz inequality.

Regarding the last non-constant term in (13), it holds

$$-\big\langle\mathbf{A}_t\mathbf{S}_t\Delta\mathbf{B}_t^\top + \Delta\mathbf{A}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top, \Delta\mathbf{A}_t\Delta\mathbf{B}_t^\top\big\rangle_{\mathrm{F}}$$

$$
\begin{aligned}
&= \eta^3 \big\langle \mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top \nabla\ell(\mathbf{W}_t) + \nabla\ell(\mathbf{W}_t)\mathbf{B}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top, \nabla\ell(\mathbf{W}_t)\mathbf{B}_t\mathbf{A}_t^\top\nabla\ell(\mathbf{W}_t)\big\rangle_{\mathrm{F}} \\
&\le \eta^3 \big(\big\|\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top\nabla\ell(\mathbf{W}_t)\big\|_{\mathrm{F}} + \big\|\nabla\ell(\mathbf{W}_t)\mathbf{B}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\big\|_{\mathrm{F}}\big)\big\|\nabla\ell(\mathbf{W}_t)\mathbf{B}_t\mathbf{A}_t^\top\nabla\ell(\mathbf{W}_t)\big\|_{\mathrm{F}} \\
&\le \eta^3\|\nabla\ell(\mathbf{W}_t)\|_2^3\big(\big\|\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top\big\|_{\mathrm{F}} + \big\|\mathbf{B}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\big\|_{\mathrm{F}}\big)\|\mathbf{B}_t\mathbf{A}_t^\top\|_{\mathrm{F}} = \mathcal{O}(\eta^3). \qquad (17)
\end{aligned}
$$

When performing fine-tuning from a pre-trained weight, both η and $\|\nabla\ell(\mathbf{W}_t)\|_2$ are observed to be tiny[1]. As a result, (17) is dominated by (16), and is neglectable in practice; see also [59, 66].

Plugging (14)-(17) into (13) yields

$$
\begin{aligned}
\ell(\mathbf{W}_t + \Delta\tilde{\mathbf{W}}_t(\mathbf{S}_t)) \le{}& \frac{L\eta^2}{2}\|\nabla\ell(\mathbf{W}_t)\|_2^2\Big[\Big\|\frac{1}{L\eta}\mathbf{I}_m - \mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top\Big\|_{\mathrm{F}}^2 + \Big\|\frac{1}{L\eta}\mathbf{I}_n - \mathbf{B}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\Big\|_{\mathrm{F}}^2 + \\
&2\big\|\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top\big\|_{\mathrm{F}}\big\|\mathbf{B}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\big\|_{\mathrm{F}}\Big] + \mathcal{O}(L\eta^3) + \mathrm{Const.} \\
\overset{(a)}{=}{}& \frac{L\eta^2}{2}\|\nabla\ell(\mathbf{W}_t)\|_2^2\Big[\big\|\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top\big\|_{\mathrm{F}}^2 - \frac{2}{L\eta}\big\langle\mathbf{I}_m, \mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top\big\rangle_{\mathrm{F}} + \big\|\mathbf{B}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\big\|_{\mathrm{F}}^2 - \\
&\frac{2}{L\eta}\big\langle\mathbf{I}_n, \mathbf{B}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\big\rangle_{\mathrm{F}} + 2\big\|\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top\big\|_{\mathrm{F}}\big\|\mathbf{B}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\big\|_{\mathrm{F}}\Big] + \mathcal{O}(L\eta^3) + \mathrm{Const.} \\
\overset{(b)}{=}{}& \frac{L\eta^2}{2}\|\nabla\ell(\mathbf{W}_t)\|_2^2\Big[\big(\big\|\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top\big\|_{\mathrm{F}} + \big\|\mathbf{B}_t\mathbf{S}_t^{-1}\mathbf{B}_t^\top\big\|_{\mathrm{F}}\big)^2 - \frac{2}{L\eta}\times \\
&\big(\|\mathbf{A}_t\mathbf{S}_t^{\frac{1}{2}}\|_{\mathrm{F}}^2 + \|\mathbf{B}_t\mathbf{S}_t^{-\frac{1}{2}}\|_{\mathrm{F}}^2\big)\Big] + \mathcal{O}(L\eta^3) + \mathrm{Const.} \\
\overset{(c)}{\le}{}& \frac{L\eta^2}{2}\|\nabla\ell(\mathbf{W}_t)\|_2^2\Big[\big(\big\|\mathbf{A}_t\mathbf{S}_t^{\frac{1}{2}}\big\|_{\mathrm{F}}^2 + \big\|\mathbf{B}_t\mathbf{S}_t^{-\frac{1}{2}}\big\|_{\mathrm{F}}^2\big)^2 - \frac{2}{L\eta}\times \\
&\big(\|\mathbf{A}_t\mathbf{S}_t^{\frac{1}{2}}\|_{\mathrm{F}}^2 + \|\mathbf{B}_t\mathbf{S}_t^{-\frac{1}{2}}\|_{\mathrm{F}}^2\big)\Big] + \mathcal{O}(L\eta^3) + \mathrm{Const.} \\
\overset{(d)}{=}{}& \frac{L\eta^2}{2}\|\nabla\ell(\mathbf{W}_t)\|_2^2\Big(\big\|\mathbf{A}_t\mathbf{S}_t^{\frac{1}{2}}\big\|_{\mathrm{F}}^2 + \big\|\mathbf{B}_t\mathbf{S}_t^{-\frac{1}{2}}\big\|_{\mathrm{F}}^2 - \frac{1}{L\eta}\Big)^2 + \mathcal{O}(L\eta^3) + \mathrm{Const.}
\end{aligned}
$$

where (a) expands the squared Frobenius norm; (b) follows from $\langle\mathbf{I}_m, \mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top\rangle_{\mathrm{F}} = \mathrm{tr}(\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top) = \|\mathbf{A}_t\mathbf{S}_t^{1/2}\|_{\mathrm{F}}^2$; (c) is because $\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top$ is SPD, thus $\|\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top\|_{\mathrm{F}} = \mathrm{tr}^{1/2}(\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top) = [\sum_i\lambda_i(\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top)]^{1/2} = [\sum_i\lambda_i^2(\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top)]^{1/2} \le \sum_i\lambda_i(\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top) = \mathrm{tr}(\mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^\top) = \|\mathbf{A}_t\mathbf{S}_t^{1/2}\|_{\mathrm{F}}^2$; and (d) utilizes completing the square. □

B.3 Proof of Theorem 3

Proof. For SPD matrix $\mathbf{S}_t$, the eigendecomposition gives $\mathbf{S}_t = \mathbf{Q}_t^S\,\mathrm{diag}(\boldsymbol{\lambda}_t^S)\mathbf{Q}_t^{S\top}$, where $\mathbf{Q}_t^S \in \mathrm{O}(r)$ and $\boldsymbol{\lambda}_t^S$ is element-wise positive. Thus, the objective (8) can be reformulated as

$$
\begin{aligned}
&\min_{\substack{\mathbf{Q}_t^S\in\mathrm{O}(r)\\ \boldsymbol{\lambda}_t^S\succeq 0}} \Big(\big\|\mathbf{A}_t\mathbf{Q}_t^S\,\mathrm{diag}^{\frac{1}{2}}(\boldsymbol{\lambda}_t^S)\mathbf{Q}_t^{S\top}\|_{\mathrm{F}}^2 + \big\|\mathbf{B}_t\mathbf{Q}_t^S\,\mathrm{diag}^{-\frac{1}{2}}(\boldsymbol{\lambda}_t^S)\mathbf{Q}_t^{S\top}\big\|_{\mathrm{F}}^2 - \frac{1}{L\eta}\Big)^2 \\
&= \min_{\substack{\mathbf{Q}_t^S\in\mathrm{O}(r)\\ \boldsymbol{\lambda}_t^S\succeq 0}} \Big(\big\|\mathbf{A}_t\mathbf{Q}_t^S\,\mathrm{diag}^{\frac{1}{2}}(\boldsymbol{\lambda}_t^S)\|_2^2 + \big\|\mathbf{B}_t\mathbf{Q}_t^S\,\mathrm{diag}^{-\frac{1}{2}}(\boldsymbol{\lambda}_t^S)\big\|_{\mathrm{F}}^2 - \frac{1}{L\eta}\Big)^2 \\
&= \min_{\substack{\mathbf{Q}_t^S\in\mathrm{O}(r)\\ \boldsymbol{\lambda}_t^S\succeq 0}} \Big(\sum_{i=1}^r[\boldsymbol{\lambda}_t^S]_i\big\|\mathbf{A}_t[\mathbf{Q}_t^S]_i\|_2^2 + \sum_{i=1}^r[\boldsymbol{\lambda}_t^S]_i^{-1}\big\|\mathbf{B}_t[\mathbf{Q}_t^S]_i\big\|_2^2 - \frac{1}{L\eta}\Big)^2 := f(\mathbf{Q}_t^S, \boldsymbol{\lambda}_t^S). \qquad (18)
\end{aligned}
$$

Since $f(\mathbf{Q}_t^S, \boldsymbol{\lambda}_t^S)$ is continuous w.r.t. both $\mathbf{Q}_t^S \in \mathrm{O}(r)$ and $\boldsymbol{\lambda}_t^S \succeq 0$, its infimum can be determined by checking the limit of f as it approaches the boundary of the open set $\mathbb{S}_{++}^r$, and analyzing stationary points in the interior [46].

Notice that $\mathrm{O}(r)$ is closed [46], while $(0, +\infty)^r$ is open. It follows from (18) that, for any given $\mathbf{Q}_t^S \in \mathrm{O}(r)$, if some $[\boldsymbol{\lambda}_t^S]_i$ approaches 0 or $+\infty$, the objective $f(\mathbf{Q}_t^S, \boldsymbol{\lambda}_t^S)$ goes to $+\infty$. As a consequence, the minimum must be attained inside the interior of $\mathbb{S}_{++}^r$.

[1]Typically, $\eta = \mathcal{O}(10^{-4})$, and $\|\nabla\ell(\mathbf{W}_t)\|_2 \le \|\nabla\ell(\mathbf{W}_t)\|_{\mathrm{F}} = \mathcal{O}(10^{-1})$ per matrix.

Next, the stationary points are investigated under the following two cases.

Case 1: $\eta \geq 1/(\tilde{C}_t L)$ or $\eta < 0$.

Defining $g(\mathbf{S}_t) := \|\mathbf{A}_t \mathbf{S}_t^{\frac{1}{2}}\|_{\mathrm{F}}^2 + \|\mathbf{B}_t \mathbf{S}_t^{-\frac{1}{2}}\|_{\mathrm{F}}^2$, it will be shown that $\min_{\mathbf{S}_t \in \mathbb{S}_{++}^r} g(\mathbf{S}_t) = \tilde{C}_t$ and the corresponding minimizer is uniquely $\tilde{\mathbf{S}}_t$.

From the stationary point condition we have

$$\begin{aligned}
&\nabla g(\mathbf{S}_t) = \mathbf{A}_t^\top \mathbf{A}_t - \mathbf{S}_t^{-1} \mathbf{B}_t^\top \mathbf{B}_t \mathbf{S}_t^{-1} = \mathbf{0} \\
\Rightarrow\ & \mathbf{S}_t \mathbf{A}_t^\top \mathbf{A}_t \mathbf{S}_t = \mathbf{B}_t^\top \mathbf{B}_t \\
\Rightarrow\ & (\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} \mathbf{S}_t (\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} (\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} \mathbf{S}_t (\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} = (\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} \mathbf{B}_t^\top \mathbf{B}_t^\top (\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} \\
\Rightarrow\ & \left[(\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} \mathbf{S}_t (\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}}\right]^2 = (\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} \mathbf{B}_t^\top \mathbf{B}_t^\top (\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}}.
\end{aligned} \tag{19}$$

Since $(\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} \mathbf{S}_t (\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}}$ is also SPD when $\mathbf{S}_t \in \mathbb{S}_{++}^r$ and $\mathrm{rank}(\mathbf{A}_t) = r$, its solution is uniquely given by the positive square root

$$(\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} \mathbf{S}_t (\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} = \left[(\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} \mathbf{B}_t^\top \mathbf{B}_t^\top (\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}}\right]^{\frac{1}{2}}.$$

Left- and right-multiplying $(\mathbf{A}_t^\top \mathbf{A}_t)^{-1/2}$ results in

$$\mathbf{S}_t = (\mathbf{A}_t^\top \mathbf{A}_t)^{-\frac{1}{2}} \left[(\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} \mathbf{B}_t^\top \mathbf{B}_t (\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}}\right]^{\frac{1}{2}} (\mathbf{A}_t^\top \mathbf{A}_t)^{-\frac{1}{2}} = \tilde{\mathbf{S}}_t.$$

Given that $g(\mathbf{S}_t)$ approaches $+\infty$ on the boundary of $\mathbb{S}_{++}^r$ and has only one stationary point $\tilde{\mathbf{S}}_t$, its minimum is reached uniquely at $\tilde{\mathbf{S}}_t$, i.e.,

$$\begin{aligned}
\min_{\mathbf{S}_t \in \mathbb{S}_{++}^r} g(\mathbf{S}_t) &= \|\mathbf{A}_t \tilde{\mathbf{S}}_t^{\frac{1}{2}}\|_{\mathrm{F}}^2 + \|\mathbf{B}_t \tilde{\mathbf{S}}_t^{-\frac{1}{2}}\|_{\mathrm{F}}^2 = \mathrm{tr}(\mathbf{A}_t \tilde{\mathbf{S}}_t \mathbf{A}_t^\top) + \mathrm{tr}(\mathbf{B}_t \tilde{\mathbf{S}}_t^{-1} \mathbf{B}_t^\top) \\
&\overset{(a)}{=} \mathrm{tr}(\mathbf{A}_t^\top \mathbf{A}_t \tilde{\mathbf{S}}_t) + \mathrm{tr}(\tilde{\mathbf{S}}_t^{-1} \mathbf{B}_t^\top \mathbf{B}_t) \overset{(b)}{=} 2\,\mathrm{tr}(\mathbf{A}_t^\top \mathbf{A}_t \tilde{\mathbf{S}}_t) \\
&= 2\|\mathbf{A}_t \tilde{\mathbf{S}}_t^{\frac{1}{2}}\|_{\mathrm{F}}^2 = 2\|\mathbf{B}_t \tilde{\mathbf{S}}_t^{-\frac{1}{2}}\|_{\mathrm{F}}^2
\end{aligned} \tag{20}$$

where (a) relies on the cyclic property of trace, and (b) utilizes (19).

Next, we prove the minimum value in (20) can be equivalently expressed as $\|\mathbf{A}_t \mathbf{B}_t^\top\|_*$.

Let $\mathbf{A}_t = \mathbf{U}_t^A \mathbf{\Sigma}_t^A \mathbf{V}_t^{A\top}$ be the economy-sized SVD. By the definition of $\tilde{\mathbf{S}}_t$, it follows that

$$\begin{aligned}
\|\mathbf{A}_t \tilde{\mathbf{S}}_t^{\frac{1}{2}}\|_{\mathrm{F}}^2 &= \mathrm{tr}\left(\mathbf{A}_t (\mathbf{A}_t^\top \mathbf{A}_t)^{-\frac{1}{2}} \left[(\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} \mathbf{B}_t^\top \mathbf{B}_t (\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}}\right]^{\frac{1}{2}} (\mathbf{A}_t^\top \mathbf{A}_t)^{-\frac{1}{2}} \mathbf{A}_t^\top\right) \\
&\overset{(a)}{=} \mathrm{tr}\left(\left[(\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} \mathbf{B}_t^\top \mathbf{B}_t (\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}}\right]^{\frac{1}{2}}\right) = \sum_{i=1}^r \lambda_i\left(\left[(\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} \mathbf{B}_t^\top \mathbf{B}_t (\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}}\right]^{\frac{1}{2}}\right) \\
&= \sum_{i=1}^r \lambda_i^{\frac{1}{2}}\left((\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} \mathbf{B}_t^\top \mathbf{B}_t (\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}}\right) = \sum_{i=1}^r \sigma_i\left((\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} \mathbf{B}_t^\top\right) \\
&= \sum_{i=1}^r \sigma_i\left(\mathbf{V}_t^A \mathbf{\Sigma}_t^A \mathbf{V}_t^{A\top} \mathbf{B}_t^\top\right) = \sum_{i=1}^r \sigma_i\left(\mathbf{U}_t^A \mathbf{\Sigma}_t^A \mathbf{V}_t^{A\top} \mathbf{B}_t^\top\right) \\
&= \sum_{i=1}^r \sigma_i\left(\mathbf{A}_t \mathbf{B}_t^\top\right) \overset{(b)}{=} \|\mathbf{A}_t \mathbf{B}_t^\top\|_*
\end{aligned}$$

where (a) relies on the cyclic property of trace, and (b) is from the definition of nuclear norm.

From the condition $\eta \in (-\infty, 0) \cup (1/(\tilde{C}_t L), +\infty)$, we have $g(\mathbf{S}_t) - 1/(L\eta) \geq 0,\ \forall \mathbf{S}_t \in \mathbb{S}_{++}^r$, thus

$$\arg\min_{\mathbf{S}_t \in \mathbb{S}_{++}^r} \left(\|\mathbf{A}_t \mathbf{S}_t^{\frac{1}{2}}\|_{\mathrm{F}}^2 + \|\mathbf{B}_t \mathbf{S}_t^{-\frac{1}{2}}\|_{\mathrm{F}}^2 - \frac{1}{L\eta}\right)^2 = \arg\min_{\mathbf{S}_t \in \mathbb{S}_{++}^r} \left(g(\mathbf{S}_t) - \frac{1}{L\eta}\right)^2 = \arg\min_{\mathbf{S}_t \in \mathbb{S}_{++}^r} g(\mathbf{S}_t) - \frac{1}{L\eta} = \tilde{\mathbf{S}}_t.$$

Case 2: $0 < \eta < 1/(\tilde{C}_t L)$.

For this case, it holds $\min_{\mathbf{S}_t \in \mathbb{S}^r_{++}} g(\mathbf{S}_t) - 1/(L\eta) < 0$, hence the minimum of (8) is reached when $g(\mathbf{S}_t) = 1/(L\eta)$. In particular, for any $\mathbf{S}_t$ satisfying $g(\mathbf{S}_t) < 1/(L\eta)$, it is possible to find a scaling factor $\gamma_t > 0$ by solving a quadratic equation such that $g(\gamma_t \mathbf{S}_t) = 1/(L\eta)$.

As an example, we consider scaling $\tilde{\mathbf{S}}_t$ to attain $g(\gamma_t \tilde{\mathbf{S}}_t) = 1/(L\eta)$. It is worth emphasizing that, for any choice $\eta \in (0, 1/(\tilde{C}_t L))$, this solution is universally valid because $g(\tilde{\mathbf{S}}_t) < 1/(L\eta)$ always hold in this case.

By the definition of $g(\mathbf{S}_t)$, the sought $g(\gamma_t \tilde{\mathbf{S}}_t) = 1/(L\eta)$ is equivalent to the quadratic equation

$$\frac{1}{L\eta} = \gamma_t \|\mathbf{A}_t \tilde{\mathbf{S}}_t^{\frac{1}{2}}\|_{\mathrm{F}}^2 + \gamma_t^{-1} \|\mathbf{B}_t \mathbf{S}_t^{-\frac{1}{2}}\|_{\mathrm{F}}^2, \ \gamma_t > 0.$$

Solving this equation gives

$$\gamma_t = \frac{\frac{1}{L\eta} \pm \sqrt{\frac{1}{L^2\eta^2} - 4\|\mathbf{A}_t \tilde{\mathbf{S}}_t^{\frac{1}{2}}\|_{\mathrm{F}}^2 \|\mathbf{B}_t \tilde{\mathbf{S}}_t^{-\frac{1}{2}}\|_{\mathrm{F}}^2}}{2\|\mathbf{A}_t \tilde{\mathbf{S}}_t^{\frac{1}{2}}\|_{\mathrm{F}}^2} = \frac{1}{\tilde{C}_t L\eta} \pm \sqrt{\frac{1}{\tilde{C}_t^2 L^2 \eta^2} - 1}$$

which concludes the proof. □

B.4 Proof of Theorem 5

Proof. Plugging $\mathbf{S}_t = s_t \mathbf{I}_r, \ s_t \in \mathbb{R}_{++}$ into (8) incurs alternative objective

$$\min_{s_t \in \mathbb{R}_{++}} h(s_t) := \Big(\|\mathbf{A}_t\|_{\mathrm{F}}^2 s_t + \|\mathbf{B}_t\|_{\mathrm{F}}^2 s_t^{-1} - \frac{1}{L\eta} \Big)^2. \tag{21}$$

To solve this quartic optimization problem, we consider the following two cases.

Case 1: $\eta \geq 1/(2\|\mathbf{A}_t\|_{\mathrm{F}} \|\mathbf{B}_t\|_{\mathrm{F}} L)$ or $\eta < 0$.

In this case we have

$$\|\mathbf{A}_t\|_{\mathrm{F}}^2 s_t + \|\mathbf{B}_t\|_{\mathrm{F}}^2 s_t^{-1} \geq 2\|\mathbf{A}_t\|_{\mathrm{F}} \|\mathbf{B}_t\|_{\mathrm{F}} \geq \frac{1}{L\eta}.$$

Since $(\cdot)^2$ monotonically increases on $\mathbb{R}_+$, the unique global optimum of (21) is

$$s_t^* = \operatorname*{arg\,min}_{s_t \in \mathbb{R}_{++}} \|\mathbf{A}_t\|_{\mathrm{F}}^2 s_t + \|\mathbf{B}_t\|_{\mathrm{F}}^2 s_t^{-1} - \frac{1}{L\eta} = \frac{\|\mathbf{B}_t\|_{\mathrm{F}}}{\|\mathbf{A}_t\|_{\mathrm{F}}}.$$

Case 2: $0 < \eta < 1/(2\|\mathbf{A}_t\|_{\mathrm{F}} \|\mathbf{B}_t\|_{\mathrm{F}} L)$.

The proof for this case relies on Descartes' rule of signs; cf. Theorem 8.

The gradients of the objective function h is given by

$$\begin{aligned} h'(s_t) &= 2\|\mathbf{A}_t\|_{\mathrm{F}}^4 s_t - 2\|\mathbf{B}_t\|_{\mathrm{F}}^4 s_t^{-3} - \frac{2}{L\eta}\|\mathbf{A}_t\|_{\mathrm{F}}^2 + \frac{2}{L\eta}\|\mathbf{B}_t\|_{\mathrm{F}}^2 s_t^{-2} \\ &= 2s_t^{-3}\Big(\|\mathbf{A}_t\|_{\mathrm{F}}^4 s_t^4 - \frac{1}{L\eta}\|\mathbf{A}_t\|_{\mathrm{F}}^2 s_t^3 + \frac{2}{L\eta}\|\mathbf{B}_t\|_{\mathrm{F}}^2 s_t - 2\|\mathbf{B}_t\|_{\mathrm{F}}^4 \Big) \end{aligned} \tag{22}$$

where the quartic polynomial in the parenthesis has coefficients with signs $(+, -, +, -)$. Using Theorem 8, the gradient (22) has 3 or 1 positive roots. That says, (21) has either 3 or 1 stationary point(s).

Notice that the objective (21) must be non-negative, and its lower bound of 0 can be reached if and only if $\|\mathbf{A}_t\|_{\mathrm{F}}^2 s_t + \|\mathbf{B}_t\|_{\mathrm{F}}^2 s_t^{-1} - 1/(L\eta) = 0$. Solving this quadratic equation over $s_t \in \mathbb{R}_{++}$ leads to two global minimum

$$s_t^* = \frac{\frac{1}{L\eta} \pm \sqrt{\frac{1}{L^2\eta^2} - 4\|\mathbf{A}_t\|_{\mathrm{F}}^2 \|\mathbf{B}_t\|_{\mathrm{F}}^2}}{2\|\mathbf{A}_t\|_{\mathrm{F}}^2}. \tag{23}$$

Hence, (21) must have 3 stationary points. As $\min_{s_t > 0} \|\mathbf{A}_t\|_{\mathrm{F}}^2 s_t + \|\mathbf{B}_t\|_{\mathrm{F}}^2 s_t^{-1} = 2\|\mathbf{A}_t\|_{\mathrm{F}} \|\mathbf{B}_t\|_{\mathrm{F}} < 1/(L\eta)$, the remaining stationary point is a local maximum at $s_t = \|\mathbf{B}_t\|_{\mathrm{F}} / \|\mathbf{A}_t\|_{\mathrm{F}}$.

Lastly, the objective (21) is a continuous function of $s_t \in (0, +\infty)$, and tends to $+\infty$ when $s_t \to 0$ and $s_t \to +\infty$. We conclude it only has two global minimum given by (23). □

B.5 Proof of Theorem 4

Proof. By the definition (5) of $\tilde{\mathbf{A}}_t$ and $\tilde{\mathbf{B}}_t$, it holds

$$\begin{aligned}\big\langle \nabla_{\tilde{\mathbf{A}}_t}\ell(\tilde{\mathbf{W}}_t), \Delta\tilde{\mathbf{A}}_t\big\rangle_{\mathrm{F}} &= \big\langle \nabla\ell(\mathbf{W}_t)\tilde{\mathbf{B}}_t, -\eta\nabla\ell(\mathbf{W}_t)\tilde{\mathbf{B}}_t\big\rangle_{\mathrm{F}} = \eta\big\langle \nabla\ell(\mathbf{W}_t)\tilde{\mathbf{B}}_t, \nabla\ell(\mathbf{W}_t)\tilde{\mathbf{B}}_t\big\rangle_{\mathrm{F}} \\ &= -\eta\big\|\nabla\ell(\mathbf{W}_t)\mathbf{B}_t\mathbf{S}_t^{-\frac{1}{2}}\big\|_{\mathrm{F}}^2 \le 0\end{aligned}$$

Likewise, we have

$$\big\langle \nabla_{\mathbf{B}_t}\ell(\mathbf{W}_t), \Delta\tilde{\mathbf{B}}_t\big\rangle_{\mathrm{F}} = -\eta\big\|\nabla\ell(\mathbf{W}_t)^\top\mathbf{A}_t\mathbf{S}_t^{\frac{1}{2}}\big\|_{\mathrm{F}}^2 \le 0.$$

As a consequence,

$$\begin{aligned}\big\langle \nabla_{\tilde{\mathbf{A}}_t}\ell(\tilde{\mathbf{W}}_t), \Delta\tilde{\mathbf{A}}_t\big\rangle_{\mathrm{F}} + \big\langle \nabla_{\tilde{\mathbf{B}}_t}\ell(\tilde{\mathbf{W}}_t), \Delta\tilde{\mathbf{B}}_t\big\rangle_{\mathrm{F}} &= -\eta\big(\|\nabla\ell(\mathbf{W}_t)\mathbf{B}_t\mathbf{S}_t^{-\frac{1}{2}}\|_{\mathrm{F}}^2 + \|\nabla\ell(\mathbf{W}_t)^\top\mathbf{A}_t\mathbf{S}_t^{\frac{1}{2}}\|_{\mathrm{F}}^2\big) \\ &\ge \|\nabla\ell(\mathbf{W}_t)\|_2^2\big(\|\mathbf{A}_t\mathbf{S}_t^{\frac{1}{2}}\|_{\mathrm{F}}^2 + \|\mathbf{B}_t\mathbf{S}_t^{-\frac{1}{2}}\|_{\mathrm{F}}^2\big).\end{aligned}$$

It follows from (20) that $\mathbf{S}_t = \tilde{\mathbf{S}}_t$ is the unique global minimum of this lower bound. □

B.6 Proof of Theorem 6

Proof. As $\tilde{\mathbf{A}}_t'\tilde{\mathbf{B}}_t'^\top = \mathbf{A}_t'\mathbf{B}_t'^\top = \mathbf{A}_t\mathbf{B}_t^\top = \tilde{\mathbf{A}}_t\tilde{\mathbf{B}}_t^\top$, Lemma 1 suggests there exists $\mathbf{Q}_t \in \mathrm{GL}(r)$ such that $\tilde{\mathbf{A}}_t' = \tilde{\mathbf{A}}_t\mathbf{Q}_t$ and $\tilde{\mathbf{B}}_t' = \tilde{\mathbf{B}}_t\mathbf{Q}_t^{-\top}$. Further, to prove Theorem 6, it is sufficient to prove that $\mathbf{Q}_t \in \mathrm{O}(r)$. Since $\tilde{\mathbf{A}}_t$ has full rank, the condition $\mathbf{Q}_t \in \mathrm{O}(r)$ is equivalent to $\tilde{\mathbf{A}}_t'\tilde{\mathbf{A}}_t'^\top = \tilde{\mathbf{A}}_t\tilde{\mathbf{A}}_t^\top$.

By Lemma 7, this can be simplify as

$$\tilde{\mathbf{A}}_t'\tilde{\mathbf{A}}_t'^\top = \mathbf{A}_t'\mathbf{S}_t'\mathbf{A}_t'^\top = \mathbf{A}_t'(\mathbf{A}_t'^\top\mathbf{A}_t')^{-1}(\mathbf{A}_t'^\top\mathbf{A}_t'\mathbf{B}_t'^\top\mathbf{B}_t')^{\frac{1}{2}}\mathbf{A}_t'^\top$$

Again using Lemma 1 and that $\mathbf{A}_t'\mathbf{B}_t'^\top = \mathbf{A}_t\mathbf{B}_t^\top$, there exists $\mathbf{P}_t \in \mathrm{GL}(r)$ such that $\mathbf{A}_t' = \mathbf{A}_t\mathbf{P}_t$ and $\mathbf{B}_t' = \mathbf{B}_t\mathbf{P}_t^{-\top}$, which yields

$$\begin{aligned}\tilde{\mathbf{A}}_t'\tilde{\mathbf{A}}_t'^\top &= \mathbf{A}_t\mathbf{P}_t(\mathbf{P}_t^\top\mathbf{A}_t^\top\mathbf{A}_t\mathbf{P}_t)^{-1}(\mathbf{P}_t^\top\mathbf{A}_t^\top\mathbf{A}_t\mathbf{P}_t\mathbf{P}_t^{-1}\mathbf{B}_t^\top\mathbf{B}_t\mathbf{P}_t^{-\top})^{\frac{1}{2}}\mathbf{P}_t^\top\mathbf{A}_t^\top \\ &= \mathbf{A}_t(\mathbf{A}_t^\top\mathbf{A}_t)^{-1}\mathbf{P}_t^{-\top}(\mathbf{P}_t^\top\mathbf{A}_t^\top\mathbf{A}_t\mathbf{B}_t^\top\mathbf{B}_t\mathbf{P}_t^{-\top})^{\frac{1}{2}}\mathbf{P}_t^\top\mathbf{A}_t^\top\end{aligned}$$

Notice that

$$\begin{aligned}&\big[\mathbf{P}_t^{-\top}(\mathbf{P}_t^\top\mathbf{A}_t^\top\mathbf{A}_t\mathbf{B}_t^\top\mathbf{B}_t\mathbf{P}_t^{-\top})^{\frac{1}{2}}\mathbf{P}_t^\top\big]^2 = \mathbf{P}_t^{-\top}(\mathbf{P}_t^\top\mathbf{A}_t^\top\mathbf{A}_t\mathbf{B}_t^\top\mathbf{B}_t\mathbf{P}_t^{-\top})\mathbf{P}_t^\top = \mathbf{A}_t^\top\mathbf{A}_t\mathbf{B}_t^\top\mathbf{B}_t \\ \implies\quad &\mathbf{P}_t^{-\top}(\mathbf{P}_t^\top\mathbf{A}_t^\top\mathbf{A}_t\mathbf{B}_t^\top\mathbf{B}_t\mathbf{P}_t^{-\top})^{\frac{1}{2}}\mathbf{P}_t^\top = (\mathbf{A}_t^\top\mathbf{A}_t\mathbf{B}_t^\top\mathbf{B}_t)^{\frac{1}{2}}.\end{aligned}$$

It then follows

$$\tilde{\mathbf{A}}_t'\tilde{\mathbf{A}}_t'^\top = \mathbf{A}_t(\mathbf{A}_t^\top\mathbf{A}_t)^{-1}(\mathbf{A}_t^\top\mathbf{A}_t\mathbf{B}_t^\top\mathbf{B}_t)^{\frac{1}{2}}\mathbf{A}_t^\top \overset{(a)}{=} \mathbf{A}_t\tilde{\mathbf{S}}_t\mathbf{A}_t^\top = \tilde{\mathbf{A}}_t\tilde{\mathbf{A}}_t^\top$$

where (a) applies (24) with $\mathbf{X} = \mathbf{A}_t^\top\mathbf{A}_t$ and $\mathbf{Y} = \mathbf{B}_t^\top\mathbf{B}_t$.

The proof is thereby completed. □

B.7 Useful facts

Lemma 7. *Under Assumption 1, it holds*

$$\tilde{\mathbf{S}}_t = (\mathbf{A}_t^\top\mathbf{A}_t)^{-\frac{1}{2}}\big[(\mathbf{A}_t^\top\mathbf{A}_t)^{\frac{1}{2}}\mathbf{B}_t^\top\mathbf{B}_t(\mathbf{A}_t^\top\mathbf{A}_t)^{\frac{1}{2}}\big]^{\frac{1}{2}}(\mathbf{A}_t^\top\mathbf{A}_t)^{-\frac{1}{2}} = (\mathbf{A}_t^\top\mathbf{A}_t)^{-1}\big[\mathbf{A}_t^\top\mathbf{A}_t\mathbf{B}_t^\top\mathbf{B}_t\big]^{\frac{1}{2}}.$$

Proof. It holds for any $\mathbf{X}, \mathbf{Y} \in \mathbb{S}_{++}^r$ that

$$\big[\mathbf{X}^{\frac{1}{2}}(\mathbf{X}^{\frac{1}{2}}\mathbf{Y}\mathbf{X}^{\frac{1}{2}})^{\frac{1}{2}}\mathbf{X}^{-\frac{1}{2}}\big]^2 = \mathbf{X}^{\frac{1}{2}}(\mathbf{X}^{\frac{1}{2}}\mathbf{Y}\mathbf{X}^{\frac{1}{2}})\mathbf{X}^{-\frac{1}{2}} = \mathbf{X}\mathbf{Y}.$$

Taking square root on both sides and left-multiplying $\mathbf{X}^{-1}$ give

$$\mathbf{X}^{-\frac{1}{2}}(\mathbf{X}^{\frac{1}{2}}\mathbf{Y}\mathbf{X}^{\frac{1}{2}})^{\frac{1}{2}}\mathbf{X}^{-\frac{1}{2}} = \mathbf{X}^{-1}(\mathbf{X}\mathbf{Y})^{\frac{1}{2}}. \tag{24}$$

Letting $\mathbf{X} = \mathbf{A}_t^\top\mathbf{A}_t$ and $\mathbf{Y} = \mathbf{B}_t^\top\mathbf{B}_t$ in (24) completes the proof. □

Theorem 8 (Descartes' rule of signs [10])**.** *The number of strictly positive roots (counting multiplicity) of polynomial h is equal to the number of sign changes in the coefficients of h, minus a nonnegative even number.*

C Algorithm pseudocodes

The Appendix provides the step-by-step pseudocodes for RefLoRA and its lightweight version RefLoRA-S. As our algorithms rely on Assumption 1 but LoRA intialize $\mathbf{B}_0 = \mathbf{0}$ by default, one can either utilizes other full-rank initialization schemes [40, 33, 59], or warm up the algorithm for $t_w > 0$ steps until $\mathbf{A}_t$ and $\mathbf{B}_t$ both satisfy full column rank. Without loss of generality, our pseudocodes assume $\mathbf{A}_0$ and $\mathbf{B}_0$ are already full-rank.

Algorithm 1: Refactored low-rank adaptation (RefLoRA)

Input: Loss ℓ, pre-trained weight $\mathbf{W}^{\rm pt}$, maximum iterations T, and learning rate η.
Initialize: full-rank $\mathbf{A}_0$ and $\mathbf{B}_0$.

for $t = 0, \ldots, T-1$ **do**
  Compute $\tilde{\mathbf{S}}_t = (\mathbf{A}_t^\top \mathbf{A}_t)^{-\frac{1}{2}} \left[(\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}} \mathbf{B}_t^\top \mathbf{B}_t (\mathbf{A}_t^\top \mathbf{A}_t)^{\frac{1}{2}}\right]^{\frac{1}{2}} (\mathbf{A}_t^\top \mathbf{A}_t)^{-\frac{1}{2}}$;
  if *adaptive optimizer* **then**
    Precondition gradients $\mathbf{G}_A = \nabla \ell(\mathbf{W}_t) \mathbf{B}_t \tilde{\mathbf{S}}_t^{-1},\ \mathbf{G}_B = \nabla \ell(\mathbf{W}_t)^\top \mathbf{A}_t \tilde{\mathbf{S}}_t$ where $\mathbf{W}_t := \mathbf{W}^{\rm pt} + \mathbf{A}_t \mathbf{B}_t^\top$;
    Update $\mathbf{A}_{t+1} = \text{AdaptOpt}(\mathbf{A}_t, \eta, \mathbf{G}_A, t),\ \mathbf{B}_{t+1} = \text{AdaptOpt}(\mathbf{B}_t, \eta, \mathbf{G}_B, t)$;
  else
    Refactor $\tilde{\mathbf{A}}_t = \mathbf{A}_t \tilde{\mathbf{S}}^{1/2},\ \tilde{\mathbf{B}}_t = \mathbf{B}_t \tilde{\mathbf{S}}^{-1/2}$;
    Update $\mathbf{A}_{t+1} = \tilde{\mathbf{A}}_t - \eta \nabla \ell(\mathbf{W}_t) \tilde{\mathbf{B}}_t,\ \mathbf{B}_{t+1} = \tilde{\mathbf{B}}_t - \eta \nabla \ell(\mathbf{W}_t)^\top \tilde{\mathbf{A}}_t$ where $\mathbf{W}_t := \mathbf{W}^{\rm pt} + \mathbf{A}_t \mathbf{B}_t^\top$;
  end
end

Output: $\mathbf{A}_T$ and $\mathbf{B}_T$.

Algorithm 2: Low-rank adaptation with simplified refactoring (RefLoRA-S)

Input: Loss ℓ, pre-trained weight $\mathbf{W}^{\rm pt}$, maximum iterations T, and learning rate η.
Initialize: full-rank $\mathbf{A}_0$ and $\mathbf{B}_0$.

for $t = 0, \ldots, T-1$ **do**
  Compute $\tilde{s}_t = \|\mathbf{B}_t\|_{\rm F} / \|\mathbf{A}_t\|_{\rm F}$;
  Refactor $\tilde{\mathbf{A}}_t = \sqrt{\tilde{s}} \mathbf{A}_t,\ \tilde{\mathbf{B}}_t = 1/\sqrt{\tilde{s}} \mathbf{B}_t$;
  if *adaptive optimizer* **then**
    Update $\mathbf{A}_{t+1} = \text{AdaptOpt}(\tilde{\mathbf{A}}_t, \eta, \nabla \ell(\mathbf{W}_t) \tilde{\mathbf{B}}_t, t)$,
    $\mathbf{B}_{t+1} = \text{AdaptOpt}(\mathbf{B}_t, \eta, \nabla \ell(\mathbf{W}_t)^\top \tilde{\mathbf{A}}_t, t)$ where $\mathbf{W}_t := \mathbf{W}^{\rm pt} + \mathbf{A}_t \mathbf{B}_t^\top$;
  else
    Update $\mathbf{A}_{t+1} = \tilde{\mathbf{A}}_t - \eta \nabla \ell(\mathbf{W}_t) \tilde{\mathbf{B}}_t,\ \tilde{\mathbf{B}}_{t+1} = \tilde{\mathbf{B}}_t - \eta \nabla \ell(\mathbf{W}_t)^\top \tilde{\mathbf{A}}_t$ where $\mathbf{W}_t := \mathbf{W}^{\rm pt} + \mathbf{A}_t \mathbf{B}_t^\top$;
  end
end

Output: $\mathbf{A}_T$ and $\mathbf{B}_T$.

D Experimental setups and hyperparameters

This appendix provides the detailed setups as well as hyperparameters used in our numerical tests.

D.1 Platforms

All the experiments are conducted on a desktop equipped with an NVIDIA RTX A5000 GPU, and a server with NVIDIA A40 and A100 GPUs. The codes for synthetical tests are written with MATLAB, and codes for LLM-related experiments are in PyTorch. In addition, our implementation of LLM tests are based on [23, 69].

D.2 Setups for visualization in Figure 1

Figure 1 considers linear regression

$$\min_{\mathbf{W}} \|\mathbf{Y} - \mathbf{W}\mathbf{X}\|_{\mathrm{F}}^2$$

where $\mathbf{X} \in \mathbb{R}^{n\times k}$ and $\mathbf{Y} \in \mathbb{R}^{m\times k}$ are random data matrices with entries generated from standard Gaussian distribution $\mathcal{N}(0,1)$. The corresponding LoRA objective is

$$\min_{\mathbf{A},\mathbf{B}} \|\mathbf{Y} - (\mathbf{W}^{\mathrm{pt}} + \mathbf{A}\mathbf{B}^\top)\mathbf{X}\|_{\mathrm{F}}^2. \tag{25}$$

For simplicity, we set $m = n = k = 2$, $r = 1$, and $\mathbf{W}^{\mathrm{pt}} = \mathbf{0}$. $\mathbf{A}_0$ and $\mathbf{B}_0$ are randomly initialized from $\mathcal{N}(0, 10)$ and $\mathcal{N}(0, \frac{1}{10})$.

Although it is well-known that linear regression has a closed-form solution through least squares, we consider it here because its Lipschitz smoothness constant can be computed analytically as $L = \|\mathbf{X}\mathbf{X}^\top\|_2$, allowing us to track the upper bound (7).

D.3 Setups for matrix factorization

Matrix factorization aims to solve

$$\min_{\mathbf{A},\mathbf{B}} \frac{1}{2}\|\mathbf{Y} - \mathbf{A}\mathbf{B}\|_{\mathrm{F}}^2$$

where $\mathbf{Y} \in \mathbb{R}^{m\times n}$ is a given low-rank matrix. It can be viewed as a special case of the linear model (25) with $k = n$, $\mathbf{X} = \mathbf{I}_n$, and $\mathbf{W}^{\mathrm{pt}} = \mathbf{0}$. This test utilizes $m = 128$, $n = 100$, and $r = 8$. The low-rank matrix $\mathbf{Y}$ is generated from standard Gaussian $\mathcal{N}(0,1)$, and then truncated to the largest r singular values. Following the standard LoRA initialization, $\mathbf{A}_0$ is sampled from standard Gaussian $\mathcal{N}(0,1)$, and $\mathbf{B}_0 = \mathbf{0}$. Standard GD is employed in LoRA for $\forall t$, while RefLoRA and ScaledGD are applied when $t > 0$.

D.4 Details of Datasets

Our evaluations are carried out on commonly-used datasets in the literature.

GLUE benchmark. The General Language Understanding Evaluation (GLUE) benchmark is designed to provide a general-purpose evaluation of natural language understanding (NLU) [58]. Those adopted in our work include

- **MNLI** [62] (Multi-Genre Natural Language Inference) tests a model's ability to perform natural language *inference* across different genres of text.
- **SST-2** [52] (Stanford Sentiment Treebank) is a *sentiment analysis* dataset with binary labels.
- **MRPC** [12] (Microsoft Research Paraphrase Corpus) focuses on *paraphrase detection*; i.e. identifying whether two sentences are semantically equivalent.
- **CoLA** [61] (Corpus of Linguistic Acceptability) requires models to judge whether a sentence is *linguistically acceptable*.
- **QNLI** [43] (Question Natural Language Inference) is a question-answering dataset converted to a binary *inference* task.
- **QQP**[2] (Quora Question Pairs) contains pairs of *questions* and the task is to determine if they are semantically equivalent.
- **RTE**[3] (Recognizing Textual Entailment) consists of sentence pairs for textual entailment *inference*.
- **STS-B** [6] (Semantic Textual Similarity Benchmark) evaluates the *textual similarity* of sentence pairs on a continuous scale.

These datasets present a comprehensive benchmark to test general-purpose language models and are distributed under various permissive licenses. A summary of these datasets can be found in Table 5.

[2] `https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs`

[3] `https://paperswithcode.com/dataset/rte`

Table 5: Summary of GLUE benchmark datasets

Dataset	Task type	|Train|	|Test|	Metric(s)
MNLI	Natural language inference	393k	20k	Matched & mismatched accuracies
SST-2	Sentiment analysis	67k	1.8k	Accuracy
MRPC	Paraphrase detection	3.7k	1.7k	Accuracy, F1
CoLA	Acceptability judgment	8.5k	1k	Matthews correlation
QNLI	QA/NLI	105k	5.4k	Accuracy
QQP	Paraphrase detection	364k	391k	Accuracy, F1
RTE	Textual entailment	2.5k	3k	Accuracy
STS-B	Semantic similarity	7k	1.4k	Pearson & Spearman Correlations

Commonsense reasoning. This category includes tasks that require models to apply everyday knowledge and infer beyond explicit textual information. These datasets are vital for evaluating a model's ability to reason about physical and social contexts. The considered datasets include

- **BoolQ** [9] (Boolean Questions) is a reading comprehension dataset of yes/no questions paired with Wikipedia passages, testing a model's ability to extract and reason over text.
- **WG** [48] (WinoGrande) is a challenging dataset designed to reduce annotation artifacts present in traditional Winograd schemas.
- **PIQA** [4] (Physical Interaction QA) assesses knowledge of physical commonsense and intuitive physics.
- **SIQA** [49] (SOCIAL-I-QA) focuses on social interaction and social commonsense reasoning.
- **HS** [67] (HellaSwag) aims at evaluating grounded commonsense inference for multiple-choice sentence completion.
- **ARC** [8] (AI2 Reasoning Challenge) contains grade-school level science questions, split into **ARCe** (ARC-easy) and **ARCc** (ARC-challenge), based on difficulty.
- **OpenbookQA** [41] involves multiple-choice science questions that require integrating commonsense and scientific facts.

These datasets are drawn from multiple domains and present diverse reasoning challenges. All datasets used in our work are publicly available under open or research-friendly licenses. Table 6 summarizes these datasets.

Table 6: Summary of commonsense reasoning datasets

Dataset	Task type	|Train|	|Test|	Metric
WinoGrande	Coreference resolution	40k	1.3k	Accuracy
PIQA	Physical reasoning	16k	3k	Accuracy
SIQA	Social reasoning	33k	2k	Accuracy
HellaSwag	Sentence completion	70k	10k	Accuracy
ARC-easy	Multiple choice QA	2.3k	1.2k	Accuracy
ARC-challenge	Multiple choice QA	2.6k	1.2k	Accuracy
OpenbookQA	Open-book QA	5.0k	500	Accuracy

D.5 Details on LLMs

We summarize the adopted language models in our evaluation. All model checkpoints are obtained from HuggingFace.

DeBERTaV3-base [20] is a transformer-based language model with 184 million parameters. The model checkpoint[4] is released under the MIT license.

[4] `https://huggingface.co/microsoft/deberta-v3-base`

GPT3-turbo is a proprietary language model accessible via the OpenAI API. While the model weights are not publicly available, its tokenizer[5] is open-sourced under the MIT license.

LLaMA-7B [56] is a decoder-only transformer model, which is part of the LLaMA (Large Language Model Meta AI) series. The chekpoint[6] is intended for research use under a non-commercial license.

LLaMA2-7B [57] is a refined successor to LLaMA. Its checkpoint[7] is under a permissive license for both research and commercial use.

LLaMA3-8B [16] is part of the third generation of LLaMA series. The checkpoint[8] is released under a permissive Meta license for both research and commercial applications.

Stable Diffusion V1.4 [44] is a latent text-to-image diffusion model released by CompVis, Stability AI, and Runway. The checkpoint[9] is made available under the CreativeML-OpenRAIL-M license.

D.6 Hyperparameters for fine-tuning LLMs

GLUE setup follows from [22, 69], where LoRA is applied to all linear modules. The results for Full FT, BitFit, Adapters, LoRA, DoRA and AdaLoRA in Table 2 are taken from [69], while the remaining results are obtained from our experiments. We fix the LoRA rank as $r = 8$ and scaling factor as $\alpha = 8$, and search the optimal learning rate from $\eta \in \{1 \times 10^{-3}, 8 \times 10^{-4}, 4 \times 10^{-4}\}$. Batch size is fixed as 32 for all datasets. The default AdamW [39] optimizer and linear learning rate scheduler are utilized for all tests. Other hyperparameters are gathered in Table 7. Note that RefLoRA requires less fine-tuning epochs compared other LoRA variants due to its fast convergence.

Table 7: Hyperparameters for GLUE benchmark

Dataset	η	Epochs	Warmup steps	Max seq. len.	Cls. dropout	Weight decay
MNLI	4×10^{-4}	5	1000	256	0.15	0
SST-2	1×10^{-3}	2	500	128	0	0.01
MRPC	4×10^{-4}	10	50	128	0	0.01
CoLA	1×10^{-3}	5	100	64	0.15	0
QNLI	4×10^{-4}	5	500	512	0.1	0.01
QQP	1×10^{-3}	5	1000	320	0.2	0.01
RTE	8×10^{-4}	10	50	320	0.2	0.01
STS-B	1×10^{-3}	10	100	128	0.1	0.1

Commonsense reasoning setup is from [23, 64], where LoRA is attached to linear projections in transformers' self-attention and feedforward modules. The results for ChatGPT, LoRA, and DoRA in Table 3 are taken from [64], while the remaining are acquired through our tests. We test with LoRA ranks $r \in \{16, 32\}$ and scaling factor fixed as $\alpha = 2r$. The learning rate is tuned from $\eta \in \{8 \times 10^{-5}, 1 \times 10^{-4}, 2 \times 10^{-4}, 3 \times 10^{-4}\}$. The tuned learning rate per model can be found in Table 8. The number of fine-tuning epochs is set to 2, and batch size is 16 for all tests. The default AdamW [39] optimizer and linear learning rate scheduler are utilized with 100 warmup steps. Dropout rate is 0.05. The remaining hyperparameters are set to the default values used in [23].

Table 8: Learning rates for LLaMA models

	LLaMA-7B		**LLaMA2-7B**		**LLaMA3-8B**	
Rank r	16	32	16	32	16	32
Learning rate η	2×10^{-4}	2×10^{-4}	3×10^{-4}	2×10^{-4}	8×10^{-5}	1×10^{-4}

[5] `https://huggingface.co/Xenova/gpt-3.5-turbo`
[6] `https://huggingface.co/huggyllama/llama-7b`
[7] `https://huggingface.co/meta-llama/Llama-2-7b`
[8] `https://huggingface.co/meta-llama/Meta-Llama-3-8B`
[9] `https://huggingface.co/CompVis/stable-diffusion-v1-4`

We also attempted to include LoRA-Pro [60]; however, it incurred runtime costs that exceeded the limits of our resources. Similar scalability concerns have been reported by other users in the community; see e.g., issue #6 on the LoRA-Pro GitHub repository. For this reasons, we omitted it from our final results.

Subject-driven image generation utilizes the default setups in [47]. Specifically, LoRA is applied to the `to_k`, `to_q`, `to_v`, `to_out`, `add_k_proj`, and `add_v_proj` modules of U-net with rank $r = 4$, and scaling factor $\alpha = 4$. The diffusion model is fine-tuned with batch size 1 and learning rate $\eta = 10^{-4}$ for 500 iterations. AdamW with 0.01 weight decay is adopted as the optimizer.

E Scaling to larger models

Scalability does not confine the applicability of RefLoRA. In larger models, the major runtime bottleneck is the forward pass and gradient computation. In comparison, the additional overhead of RefLoRA becomes negligible. Table 9 compares the computational overheads of LoRA variants under various model sizes, where gradient checkpointing is turned on for Gemma3-27B-pt so that the model can fit within a single NVIDIA H100 96GB GPU. Notably, the runtime and memory gap between LoRA and RefLoRA narrows as model size increases. These findings indicate that RefLoRA remains as practical and efficient as LoRA, especially for larger models.

Table 9: Throughput (it/s↑) and GPU memory consumption (GB↓) under various models sizes.

Method	**DeBERTaV3-base**		**LLaMA3-8B**		**Gemma3-27B-pt**	
	Tp.	Mem.	Tp.	Mem.	Tp.	Mem.
LoRA	1× (2.26)	8.73	1× (1.53)	36.21	1× (0.58)	64.28
LoRA-RITE	0.73×	8.87	0.63×	37.37	0.95×	64.28
RefLoRA	0.88×	8.86	0.87×	36.35	0.98×	64.28
RefLoRA-S	0.99×	8.73	0.99×	36.21	0.99×	64.28

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: The main claims in the abstract and introduction are clearly stated, which are supported by both theoretical analysis in Section 3 and experiments Section 4.

   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: The limitations of RefLoRA are acknowledged in the outlook of Section 5 and future directions in Appendix A.

   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: The paper includes clearly stated Assumptions 1-2 and detailed proofs in Appendix B for all theoretical results.

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: Appendix D contains full details of datasets, hyperparameters, training setups, and platform specs.

   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: [Yes]

Justification: The models and datasets used are publicly available with links in Section D. Codes for reproducing the main results are provided as supplementary material.

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: Detailed training settings including datasets, hyperparameters, and optimizers are documented in Appendix D.

   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: [No]

   Justification: While results are averaged over multiple runs, error bars are not reported because the overall performance is computed across multiple datasets that differ in size and use heterogeneous metrics, making it improper to calculate a unified error bar. Additionally, space constraints in performance tables prevent the inclusion of error bars, and for fair comparison, we follow prior works which also do not report error bars for their methods.

   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: Appendix D.1 lists the compute resources (NVIDIA A40, A100, and RTX A5000) used for experiments. Section 4.5 compares the throughput and GPU usage.

   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 paper does not involve sensitive data, human subjects, or unethical practices, and abides by NeurIPS ethical guidelines.

   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: [Yes]

    Justification: The broader impacts of this work are explicitly discussed in Appendix A.

    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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This work does not release any new datasets or pre-trained models that could pose a risk of misuse. It solely builds on publicly available models and benchmarks for evaluation purposes.

    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: Original owners of codes, datasets, and models are explicitly mentioned via citations or urls in Section D.

    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: No new datasets or models are introduced; the work only modifies and evaluates existing ones.

    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: The paper does not involve crowdsourcing or human participants.

    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: No human subjects or participant-based studies are involved.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The paper focuses on the fine-tuning of LLMs, whose usages are clearly described throughout the paper.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.