FILO: A Unified Fisher-Geometric Framework for Low-Rank Adaptation

Ziyun Liu **Fengmiao Bian*** **Jian-Feng Cai***
Department of Mathematics
The Hong Kong University of Science and Technology
zliueq@connect.ust.hk {mafmbian,jfcai}@ust.hk

Abstract

Low-Rank Adaptation (LoRA) reparameterizes a weight matrix $\boldsymbol{W}$ as a product of two low-rank factors, but the Jacobian $J_{\mathcal{G}}$ of the generator mapping the factors to the weight matrix is rank-deficient, so the factor-space preconditioner $J_{\mathcal{G}}^{*}\mathcal{F}_t J_{\mathcal{G}}$ induced by any $\boldsymbol{W}$-space preconditioner $\mathcal{F}_t$ is singular, and consequently the standard chain rule cannot be uniquely inverted to map a preconditioned $\boldsymbol{W}$-space direction back to a factor-space update. Therefore, we introduce **FILO**, a unified Fisher-geometric framework for low-rank adaptation that enables factor updates without directly inverting this singular operator and casts existing LoRA optimizers by their factor-space inversion strategy and $\boldsymbol{W}$-space preconditioner $\mathcal{F}_t$. Existing methods occupy four families along these axes: factor-space adaptive updates, block-diagonal surrogates for $J_{\mathcal{G}}^{*}J_{\mathcal{G}}$, Frobenius-residual pseudoinverse methods, and Riemannian manifold methods. Within this design space, a gradient-statistics-aware $\mathcal{F}_t$ paired with a closed-form factor-space solve at $\mathcal{O}((m+n)r)$ memory remains underexplored. We instantiate FILO as **FILO-Adafactor** by adopting the Adafactor diagonal Kronecker preconditioner $\mathcal{H}_t$ on $\boldsymbol{W}$ and selecting from the resulting factor-space solution family the element minimizing an $\mathcal{H}_t$-weighted imbalance between the two factor contributions. By construction, the induced weight update is the closest LoRA approximation to the preconditioned $\boldsymbol{W}$-space direction under the $\mathcal{H}_t$-weighted norm. Across GPT-2 on E2E, Mistral-7B and Qwen2-7B on GLUE, ARC, and GSM8K, and diffusion-model personalization, FILO-Adafactor and its momentum variant are competitive with or improve over a representative set of LoRA optimizers while keeping peak GPU memory at the LoRA optimizer level. An open-source implementation is available at FILO.

1 Introduction

Fine-tuning large pretrained models [2, 20, 36] for downstream tasks is increasingly bottlenecked by the cost of full-parameter updates, motivating parameter-efficient fine-tuning (PEFT). Low-Rank Adaptation (LoRA) [15] has become the standard PEFT method: it freezes the pretrained weight $\boldsymbol{W}_0$ and reparameterizes its update as a product $\boldsymbol{BA}$ with $\boldsymbol{B} \in \mathbb{R}^{m\times r}$, $\boldsymbol{A} \in \mathbb{R}^{r\times n}$, $r \ll \min(m,n)$, reducing trainable parameters and optimizer state from $\mathcal{O}(mn)$ to $\mathcal{O}((m+n)r)$. A growing line of work [12, 22, 33, 34, 38–41] extends this template with refined optimizers in pursuit of full-fine-tuning quality at LoRA's memory budget.

Despite this progress, optimizing in the factor space $[\boldsymbol{B}, \boldsymbol{A}]$ rather than directly in $\boldsymbol{W}$ raises a fundamental obstruction (§ 2): writing $\mathcal{G} : [\boldsymbol{B}, \boldsymbol{A}] \mapsto \boldsymbol{BA}$ for the map from the factors to the weight matrix, its Jacobian $J_{\mathcal{G}}$ is rank-deficient because $\mathcal{G}$ has a built-in redundancy under the gauge

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

reparameterization $(\boldsymbol{B}, \boldsymbol{A}) \mapsto (\boldsymbol{BC}, \boldsymbol{C}^{-1}\boldsymbol{A})$ for any invertible $\boldsymbol{C}$. This redundancy also constrains preconditioning in factor space. Practical preconditioners are often motivated by approximations to the Fisher information in the optimized parameterization [18, 21], we therefore consider the empirical Fisher with respect to $[\boldsymbol{B}, \boldsymbol{A}]$. By the chain rule, the factor-space empirical Fisher must take the form $J_{\mathcal{G}}^{*}\mathcal{F}_t J_{\mathcal{G}}$, where $\mathcal{F}_t$ is the corresponding empirical Fisher in $\boldsymbol{W}$-space. Because this gauge redundancy makes $J_{\mathcal{G}}^{*}\mathcal{F}_t J_{\mathcal{G}}$ singular, ordinary inversion cannot be used to obtain a factor-space update from a $\boldsymbol{W}$-space preconditioned direction, and different generalized inverses can yield different factor updates. We therefore introduce **FILO**, a unified **FI**sher-geometric framework for **LO**w-rank adaptation that organizes existing LoRA optimizers along two axes: the factor-space inversion strategy and the choice of $\boldsymbol{W}$-space preconditioner $\mathcal{F}_t$. Viewed through FILO, cheap factor-space schemes preserve the $\mathcal{O}((m+n)r)$ memory budget but discard the gradient-statistics structure on $\boldsymbol{W}$. Vanilla LoRA [15] and Imbalance-Reg [41] bypass the framework altogether by applying per-coordinate adaptive updates directly on the factors. LoRA+ [12] and Riemannian Preconditioned LoRA [38] take $\mathcal{F}_t = \boldsymbol{I}$ and use block-diagonal approximations of $J_{\mathcal{G}}^{*}J_{\mathcal{G}}$. LoRA-Pro [34] fits the preconditioned weight-space direction $\mathcal{F}_t^{-1}\boldsymbol{G}_t$ by minimizing the Frobenius residual $\|J_{\mathcal{G}}[\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}] - \mathcal{F}_t^{-1}\boldsymbol{G}_t\|_F^2$, its AdamW variant pairs a non-trivial $\mathcal{F}_t$ on $\boldsymbol{W}$ with a Frobenius residual, mismatching the preconditioner's metric, and explicitly maintains $\boldsymbol{W}$-space first/second moments at $\mathcal{O}(mn)$ memory. Manifold-based methods [17], including Riemannian Muon [5] and RAdamW [4] take a Riemannian gradient step on $\mathcal{M}_r$ in the ambient $\boldsymbol{W}$-space and rely on a retraction back to the manifold. A gradient-statistics-aware $\mathcal{F}_t$ paired with a closed-form factor-space solve and $\mathcal{O}((m+n)r)$ optimizer state remains an underexplored design point.

Within FILO, we target this design point through the consistent factor-space system $J_{\mathcal{G}}^{*}\mathcal{F}_t J_{\mathcal{G}}\,[\boldsymbol{P}, \boldsymbol{Q}] = J_{\mathcal{G}}^{*}(\boldsymbol{G}_t)$ for $\mathcal{F}_t \succ 0$: when both factors have rank r, its solution set forms an r^2-dimensional affine subspace. Our instantiation of FILO therefore makes two coupled choices (§ 3): (i) which gradient-statistics-aware preconditioner $\mathcal{F}_t$ to use on $\boldsymbol{W}$, and (ii) how to select a particular element of the affine solution set. For (i), we adopt the Adafactor [29] diagonal Kronecker form $\mathcal{F}_t = \mathcal{H}_t = (\boldsymbol{L}_t \otimes \boldsymbol{R}_t)^{\frac{1}{2}}$, acting as $\mathcal{H}_t\boldsymbol{Y} = \boldsymbol{L}_t^{1/2}\boldsymbol{Y}\boldsymbol{R}_t^{1/2}$, the cheapest non-trivial $\boldsymbol{W}$-space preconditioner, requiring $\mathcal{O}(m+n)$ memory. For (ii), all elements of the affine solution set induce the same first-order $\boldsymbol{W}$-update, namely the $\mathcal{H}_t$-orthogonal projection of $\mathcal{H}_t^{-1}\boldsymbol{G}_t$ onto the tangent space $\mathbb{T}_t$, but can lead to different factor updates. We therefore select the element that minimizes the $\mathcal{H}_t$-weighted imbalance $\|\boldsymbol{\Delta}_{\boldsymbol{B}_t}\boldsymbol{A}_t - \boldsymbol{B}_t\boldsymbol{\Delta}_{\boldsymbol{A}_t}\|_{\mathcal{H}_t}^2$ between the two factor contributions to the $\boldsymbol{W}$ update. The resulting algorithm, **FILO-Adafactor**, instantiates FILO with these two choices, admits a closed-form factor update at $\mathcal{O}(r^3)$ extra cost and keeps the optimizer state at $\mathcal{O}((m+n)r)$. Consequently, with current factor gradients, FILO-Adafactor's induced $\boldsymbol{W}$-update is the closest point in $\mathbb{T}_t$ to the Adafactor-preconditioned direction $\mathcal{H}_t^{-1}\boldsymbol{G}_t$ under the $\mathcal{H}_t$-weighted norm.

Empirically (§ 4), FILO-Adafactor matches or outperforms vanilla LoRA, Scaled AdamW, LoRA-Pro AdamW, and SOAP across GPT-2 on E2E, Mistral-7B and Qwen2-7B on GLUE, ARC, and GSM8K, and diffusion-model fine-tuning, while matching Scaled AdamW's peak memory and avoiding the $\sim 2\times$ memory overhead of LoRA-Pro AdamW. Our contributions are:

- **FILO**, a unified Fisher-geometric framework for low-rank adaptation that recasts existing LoRA optimizers as instances of the consistent linear system $J_{\mathcal{G}}^{*}\mathcal{F}_t J_{\mathcal{G}}\,[\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}] = J_{\mathcal{G}}^{*}(\boldsymbol{G}_t)$, parameterized by the choice of preconditioner $\mathcal{F}_t$ and the factor-space inversion strategy (§ 2.2).
- **FILO-Adafactor**, an instantiation of FILO whose $\boldsymbol{W}$-update is the closest point in $\mathbb{T}_t$ to the Adafactor-preconditioned direction $\mathcal{H}_t^{-1}\boldsymbol{G}_t$ under the $\mathcal{H}_t$-weighted norm, recovered in closed form at $\mathcal{O}((m+n)r)$ optimizer state memory (§ 3).
- Experimental evidence that the resulting update direction is competitive with or improves over both cheap factor-space baselines and baselines based on weight-space preconditioning, including at the 7B parameter scale.

2 The FILO Framework and Related Work

This section sets up the LoRA optimization problem (§ 2.1), identifies the singular factor-space operator $J_{\mathcal{G}}^{*}\mathcal{F}_t J_{\mathcal{G}}$ that any $\boldsymbol{W}$-space preconditioner $\mathcal{F}_t$ induces and the resulting non-uniqueness of factor update, organizes existing LoRA optimizers in the FILO framework by their choice of $\mathcal{F}_t$ and

their factor-space inversion strategy of $J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}}$ (§ 2.2), and reviews adaptive preconditioner families on $\boldsymbol{W}$ (§ 2.3). Throughout the paper, calligraphic letters ($\mathcal{F}_t, \mathcal{H}_t, \widetilde{\mathcal{P}}_{\mathbb{T}_t}$) denote linear operators on $\mathbb{R}^{m\times n}$, while bold letters ($\boldsymbol{L}_t, \boldsymbol{R}_t, \boldsymbol{B}_t, \boldsymbol{A}_t, \boldsymbol{G}_t$) denote matrices, $J_{\mathcal{G}}$ maps factor-space directions to weight-space directions. A complete notation table is given in Appendix B.

2.1 LoRA Setup and Its Singular Jacobian

As a representative parameter-efficient fine-tuning method, low-rank fine-tuning freezes the pretrained weight $\boldsymbol{W}_0 \in \mathbb{R}^{m\times n}$ and assumes that the weight update $\boldsymbol{W}$ admits a low-rank factorization $\boldsymbol{W} = \boldsymbol{B}\boldsymbol{A}$ with $\boldsymbol{B} \in \mathbb{R}^{m\times r}$, $\boldsymbol{A} \in \mathbb{R}^{r\times n}$, and $r \ll \min\{m, n\}$ [15]. The fine-tuning objective is

$$\min_{\boldsymbol{B}\in\mathbb{R}^{m\times r},\, \boldsymbol{A}\in\mathbb{R}^{r\times n}} \mathcal{L}(\boldsymbol{W}_0 + \mathcal{G}([\boldsymbol{B}, \boldsymbol{A}])), \quad \text{where} \quad \mathcal{G}([\boldsymbol{B}, \boldsymbol{A}]) = \boldsymbol{B}\boldsymbol{A}.$$

Under the generator $\mathcal{G}$, the Jacobian operator $J_{\mathcal{G}}([\boldsymbol{B}_t, \boldsymbol{A}_t]) : [\mathbb{R}^{m\times r}, \mathbb{R}^{r\times n}] \to \mathbb{R}^{m\times n}$ and its adjoint $J_{\mathcal{G}}^*([\boldsymbol{B}_t, \boldsymbol{A}_t]) : \mathbb{R}^{m\times n} \to [\mathbb{R}^{m\times r}, \mathbb{R}^{r\times n}]$ act as

$$J_{\mathcal{G}}([\boldsymbol{B}_t, \boldsymbol{A}_t])[\boldsymbol{P}, \boldsymbol{Q}] = \boldsymbol{P}\boldsymbol{A}_t + \boldsymbol{B}_t\boldsymbol{Q}, \qquad J_{\mathcal{G}}^*([\boldsymbol{B}_t, \boldsymbol{A}_t])(\boldsymbol{C}) = [\boldsymbol{C}\boldsymbol{A}_t^\top,\, \boldsymbol{B}_t^\top\boldsymbol{C}], \tag{1}$$

on factor-space directions $[\boldsymbol{P}, \boldsymbol{Q}] \in [\mathbb{R}^{m\times r}, \mathbb{R}^{r\times n}]$ and $\boldsymbol{W}$-space directions $\boldsymbol{C} \in \mathbb{R}^{m\times n}$, respectively. We abbreviate $J_{\mathcal{G}} := J_{\mathcal{G}}([\boldsymbol{B}_t, \boldsymbol{A}_t])$ and $J_{\mathcal{G}}^* := J_{\mathcal{G}}^*([\boldsymbol{B}_t, \boldsymbol{A}_t])$ when the base point is clear; detailed derivations appear in Proposition C.1. The chain rule gives the factor gradients $\boldsymbol{G}_{\boldsymbol{B}_t} = \boldsymbol{G}_t\boldsymbol{A}_t^\top, \boldsymbol{G}_{\boldsymbol{A}_t} = \boldsymbol{B}_t^\top\boldsymbol{G}_t$, with $\boldsymbol{G}_t = \nabla_{\boldsymbol{W}_t}\mathcal{L}(\boldsymbol{W}_0 + \boldsymbol{W}_t)$, equivalently

$$J_{\mathcal{G}}^*(\boldsymbol{G}_t) = [\boldsymbol{G}_{\boldsymbol{B}_t}, \boldsymbol{G}_{\boldsymbol{A}_t}]. \tag{2}$$

Thus $J_{\mathcal{G}}$ is the central operator linking factor-space and $\boldsymbol{W}$-space updates, and its properties determine what factor-space optimizers can achieve.

Unfortunately, $J_{\mathcal{G}}$ has a non-trivial kernel: *the Jacobian $J_{\mathcal{G}}$ is rank-deficient.* The Jacobian formula in (1) immediately produces a family of factor-space directions that $J_{\mathcal{G}}$ maps to $\boldsymbol{0} \in \mathbb{R}^{m\times n}$: for any $\boldsymbol{X} \in \mathbb{R}^{r\times r}$,

$$J_{\mathcal{G}}[\boldsymbol{B}_t\boldsymbol{X},\, -\boldsymbol{X}\boldsymbol{A}_t] = \boldsymbol{B}_t\boldsymbol{X}\boldsymbol{A}_t - \boldsymbol{B}_t\boldsymbol{X}\boldsymbol{A}_t = \boldsymbol{0}, \tag{3}$$

so $[\boldsymbol{B}_t\boldsymbol{X}, -\boldsymbol{X}\boldsymbol{A}_t] \in \ker(J_{\mathcal{G}})$. When $\boldsymbol{B}_t$ has column rank r and $\boldsymbol{A}_t$ has row rank r, this family is the entire kernel: $\ker(J_{\mathcal{G}}) = \{[\boldsymbol{B}_t\boldsymbol{X}, -\boldsymbol{X}\boldsymbol{A}_t] : \boldsymbol{X} \in \mathbb{R}^{r\times r}\}$, an r^2-dimensional subspace, so $\operatorname{rank}(J_{\mathcal{G}}) = (m+n)r - r^2$ (Proposition C.2).

This rank deficiency also constrains the form of preconditioners in factor space. Since practical preconditioners are often motivated by approximations to the Fisher information in the optimized parameterization [18, 21], the natural preconditioner for the factors $[\boldsymbol{B}_t, \boldsymbol{A}_t]$ is the empirical Fisher formed from the per-sample factor gradients $[\boldsymbol{G}_{\boldsymbol{B}_t}^{(i)}, \boldsymbol{G}_{\boldsymbol{A}_t}^{(i)}] := [\nabla_{\boldsymbol{B}}\mathcal{L}_i(\boldsymbol{W}_t), \nabla_{\boldsymbol{A}}\mathcal{L}_i(\boldsymbol{W}_t)]$, where i indexes the N training samples and $\mathcal{L}_i$ denotes the corresponding loss. That is,

$$\mathcal{E}_t \;:=\; \frac{1}{N}\sum_{i=1}^{N} \langle [\boldsymbol{G}_{\boldsymbol{B}_t}^{(i)}, \boldsymbol{G}_{\boldsymbol{A}_t}^{(i)}],\, \cdot \rangle [\boldsymbol{G}_{\boldsymbol{B}_t}^{(i)}, \boldsymbol{G}_{\boldsymbol{A}_t}^{(i)}]$$

Adaptive optimizers use structured preconditioners: diagonal for Adam [18], rank-1 Kronecker for Adafactor [29], full Kronecker for Shampoo [11], and layerwise Kronecker for K-FAC [21]. In practice, these preconditioners are built from accumulated or exponentially averaged mini-batch gradient statistics. However, as our experiments show, AdamW performs poorly in the factorized setting. This motivates a closer look at the structure of $\mathcal{E}_t$.

Because each $\mathcal{L}_i$ depends on the factors only through $\boldsymbol{W} = \boldsymbol{B}\boldsymbol{A}$, the chain rule gives $[\boldsymbol{G}_{\boldsymbol{B}_t}^{(i)}, \boldsymbol{G}_{\boldsymbol{A}_t}^{(i)}] = J_{\mathcal{G}}^*(\boldsymbol{G}_t^{(i)})$, where $\boldsymbol{G}_t^{(i)} := \nabla_{\boldsymbol{W}}\mathcal{L}_i(\boldsymbol{W}_t)$. Hence

$$\mathcal{E}_t \;=\; \frac{1}{N}\sum_{i=1}^{N} \big\langle J_{\mathcal{G}}^*(\boldsymbol{G}_t^{(i)}),\, \cdot \big\rangle J_{\mathcal{G}}^*(\boldsymbol{G}_t^{(i)}) \;=\; J_{\mathcal{G}}^*\Big(\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{G}_t^{(i)} \langle \boldsymbol{G}_t^{(i)}, \cdot \rangle\Big) J_{\mathcal{G}} \;=\; J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}}, \tag{4}$$

where $\mathcal{F}_t := \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{G}_t^{(i)} \langle \boldsymbol{G}_t^{(i)}, \cdot \rangle$ on the right is the empirical Fisher in $\boldsymbol{W}$-space [19]. Equivalently, in matrix form,

$$\mathcal{F}_t \;=\; \frac{1}{N}\sum_{i=1}^{N} \operatorname{vec}(\boldsymbol{G}_t^{(i)}) \operatorname{vec}(\boldsymbol{G}_t^{(i)})^\top \;\in\; \mathbb{R}^{mn\times mn}. \tag{5}$$

Motivated by (4), we consider a general positive-definite weight-space preconditioner $\mathcal{F}_t$. Since $J_{\mathcal{G}}$ is rank-deficient, the induced factor-space operator $J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}}$ is necessarily singular. When $\mathcal{F}_t \succ 0$, $\ker(J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}}) = \ker(J_{\mathcal{G}})$ is non-trivial, so $J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}}$ is singular for any choice of $\mathcal{F}_t$, and the corresponding preconditioned update on $[\boldsymbol{B}_t, \boldsymbol{A}_t]$,

$$[\boldsymbol{B}_{t+1}, \boldsymbol{A}_{t+1}] = [\boldsymbol{B}_t, \boldsymbol{A}_t] - \eta_t \left(J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}}\right)^{-1} [\boldsymbol{G}_{\boldsymbol{B}_t}, \boldsymbol{G}_{\boldsymbol{A}_t}], \tag{6}$$

is ill-defined.

Obstruction 1 (non-invertibility). The operator $J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}}$ is non-invertible for any choice of $\mathcal{F}_t$, so the update (6) is ill-defined.

One natural remedy is to replace the inverse with a generalized inverse, but different choices produce different factor updates, so the ill-definedness shifts from non-existence to non-uniqueness. A canonical choice is the Moore–Penrose pseudoinverse $(J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}})^\dagger$:

$$[\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}] \;:=\; (J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}})^\dagger \, [\boldsymbol{G}_{\boldsymbol{B}_t}, \boldsymbol{G}_{\boldsymbol{A}_t}] \;\in\; [\mathbb{R}^{m \times r}, \mathbb{R}^{r \times n}].$$

By property of the Moore–Penrose pseudoinverse, $[\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}]$ is the unique minimum-Frobenius-norm element of the affine solution set

$$\left\{ [\boldsymbol{P}, \boldsymbol{Q}] \in [\mathbb{R}^{m \times r}, \mathbb{R}^{r \times n}] \;:\; J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}} \, [\boldsymbol{P}, \boldsymbol{Q}] = [\boldsymbol{G}_{\boldsymbol{B}_t}, \boldsymbol{G}_{\boldsymbol{A}_t}] \right\},$$

which is consistent since $\operatorname{range}(J_{\mathcal{G}}^*) = \operatorname{range}(J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}})$ as $\mathcal{F}_t \succ 0$, and has dimension $\dim \ker(J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}}) = \dim \ker(J_{\mathcal{G}}) = r^2$ when both $\boldsymbol{B}_t$ and $\boldsymbol{A}_t$ have rank r. Other generalized inverses of $J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}}$ correspond to different elements of this affine solution set, differing by an element of $\ker(J_{\mathcal{G}})$.

Obstruction 2 (non-uniqueness of generalized inverse). Generalized inverses of $J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}}$ are not unique, and different choices can yield factor updates differing by elements of $\ker(J_{\mathcal{G}})$. An additional selection criterion is needed to determine a unique factor update.

These observations motivate FILO, a framework that separates the choice of $\mathcal{F}_t$ on $\boldsymbol{W}$ from the inversion strategy for the induced factor-space operator. Existing LoRA optimizers in § 2.2 differ in the choice of $\mathcal{F}_t$ on $\boldsymbol{W}$ and in the rule that selects an element of this affine solution set; § 2.3 reviews the standard families of $\mathcal{F}_t$. Our method FILO-Adafactor (§ 3) instantiates this framework with the Adafactor diagonal Kronecker and an $\mathcal{H}_t$-balance criterion that selects a unique element of the affine solution set.

2.2 FILO: A Unified View of LoRA Optimizers

Although $J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}}$ is singular, the linear system

$$J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}} \, [\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}] \;=\; J_{\mathcal{G}}^*(\boldsymbol{G}_t) \tag{7}$$

in the factor update $[\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}]$ is consistent, with RHS equal to the factor-gradient pair $[\boldsymbol{G}_{\boldsymbol{B}_t}, \boldsymbol{G}_{\boldsymbol{A}_t}]$ by (2). FILO organizes existing LoRA optimizers by (i) the choice of the weight-space preconditioner $\mathcal{F}_t$, which determines the preconditioned direction $\mathcal{F}_t^{-1} \boldsymbol{G}_t$ and induces the factor-space operator $J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}}$, and (ii) which invertible surrogate for this singular operator they use to obtain factor updates. For methods without such a surrogate, we instead record their generalized-inverse construction or their strategy for computing updates without solving (7). Table 1 summarizes the resulting design space, and we walk through the main families below.

Vanilla LoRA / Imbalance-Reg / LoRA-RITE (diagonal approximation of $\mathcal{E}_t$, ignoring $J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}}$). Vanilla LoRA [15] ignores $J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}}$ and approximates the empirical Fisher $\mathcal{E}_t$ on the factor space by a per-coordinate diagonal estimated directly from the factor gradients $[\boldsymbol{G}_{\boldsymbol{B}_t}, \boldsymbol{G}_{\boldsymbol{A}_t}]$ via AdamW. Imbalance-Regularized LoRA [41] keeps the same diagonal Fisher estimate and adds a penalty $\|\boldsymbol{B}_t^\top \boldsymbol{B}_t - \boldsymbol{A}_t \boldsymbol{A}_t^\top\|_F^2$ to align the factor spectra. LoRA-RITE [37] replaces the diagonal estimate with a matrix-form second moment $\boldsymbol{V}_t \in \mathbb{R}^{r \times r}$ accumulated on the polar/QR-reparameterized factor gradients, yielding a transformation-invariant factor-space update at $\mathcal{O}(r^2)$ extra memory.

LoRA+ [12] and Riemannian Preconditioned LoRA [38] (block-diagonal surrogates for $J_{\mathcal{G}}^* J_{\mathcal{G}}$). Specializing (7) to $\mathcal{F}_t = \boldsymbol{I}$, the operator $J_{\mathcal{G}}^* J_{\mathcal{G}}$ (Proposition C.1) decomposes into block-diagonal and cross terms. LoRA+ approximates $J_{\mathcal{G}}^* J_{\mathcal{G}}$ by the block-scaling identity surrogate $\operatorname{diag}(\boldsymbol{I}_m, \lambda \boldsymbol{I}_n)$

Table 1: Existing LoRA optimizers as instances of the framework (7), grouped by how they handle the singular operator $J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}}$. Inversion strategy = how $(J_{\mathcal{G}}^* \mathcal{F}_t J_{\mathcal{G}})^{-1}$ is replaced or approximated; $\mathcal{F}_t$ = the gradient-statistics structure on $\boldsymbol{W}$ used ("—" means the method bypasses (7) and does not instantiate $\mathcal{F}_t$). Per-step cost and memory are per-layer beyond forward/backward through $\boldsymbol{BA}$ and storing $[\boldsymbol{B}_t, \boldsymbol{A}_t]$. $\boldsymbol{g} := \mathrm{vec}(\boldsymbol{G}_t) \in \mathbb{R}^{mn}$ denotes the vectorized $\boldsymbol{W}$-space gradient.

Method	**Inversion strategy**	$\mathcal{F}_t$ **on** $\boldsymbol{W}$	**Per-step cost**	**Memory**
Bypass (7) *(factor-space AdamW)*				
Vanilla LoRA [15]	—	—	$\mathcal{O}((m+n)r)$	$\mathcal{O}((m+n)r)$
LoRA-RITE [37]	—	—	$\mathcal{O}((m+n)r^2)$	$\mathcal{O}((m+n)r)$
Block-diagonal approx. of $J_{\mathcal{G}}^* J_{\mathcal{G}}$				
LoRA+ [12]	Block-scaling	$\boldsymbol{I}$	$\mathcal{O}((m+n)r)$	$\mathcal{O}((m+n)r)$
Riem. Precond. [38]	Block-diag	$\boldsymbol{I}$	$\mathcal{O}((m+n)r^2)$	$\mathcal{O}((m+n)r)$
Pseudoinverse of $J_{\mathcal{G}}$				
LoRA-Pro SGD [34]	$J_{\mathcal{G}}^{\dagger}\ \mathcal{F}_t^{-1}$	$\boldsymbol{I}$	$\mathcal{O}((m+n)r^2)$	N/A
LoRA-Pro AdamW [34]	$J_{\mathcal{G}}^{\dagger}\ \mathcal{F}_t^{-1}$	$\mathrm{diag}(\boldsymbol{g} \odot \boldsymbol{g})^{\frac{1}{2}}$	$\mathcal{O}(mnr)$	$\mathcal{O}(mn)$
Riemannian gradient descent on $\mathcal{M}_r$				
Riem. Muon [5]	RGD on $\mathcal{M}_r$	$(\boldsymbol{G}_t \boldsymbol{G}_t^\top)^{1/2}$	$\mathcal{O}((m+n)r^2)$	$\mathcal{O}((m+n)r)$
RAdamW [4]	RGD on $\mathcal{M}_r$	$\mathrm{diag}(\boldsymbol{L}_{\mathrm{Sh}} \otimes \boldsymbol{R}_{\mathrm{Sh}})^{\frac{1}{4}}$	$\mathcal{O}((m+n)r^2)$	$\mathcal{O}((m+n)r)$
FILO-Adafactor (ours)	solve (7)	Adafactor diag-Kron	$\mathcal{O}((m+n)r^2)$	$\mathcal{O}((m+n)r)$

for a fixed scalar $\lambda > 0$ and inverts it, giving asymmetric per-block scalar rescaling between the $\boldsymbol{B}_t$ and $\boldsymbol{A}_t$ updates. Riemannian Preconditioned LoRA approximates $J_{\mathcal{G}}^* J_{\mathcal{G}}$ by its block-diagonal part $\mathrm{diag}(\boldsymbol{A}_t \boldsymbol{A}_t^\top, \boldsymbol{B}_t^\top \boldsymbol{B}_t)$ and inverts it, yielding the explicit factor update $\boldsymbol{\Delta}_{\boldsymbol{B}_t} = \boldsymbol{G}_{\boldsymbol{B}_t} (\boldsymbol{A}_t \boldsymbol{A}_t^\top)^{-1}$, $\boldsymbol{\Delta}_{\boldsymbol{A}_t} = (\boldsymbol{B}_t^\top \boldsymbol{B}_t)^{-1} \boldsymbol{G}_{\boldsymbol{A}_t}$, well-defined whenever $\boldsymbol{B}_t, \boldsymbol{A}_t$ have full rank. Both updates generally differ from any element of the affine solution set of (7).

LoRA-Pro [34]. LoRA-Pro solves a different system from (7): it minimizes the Frobenius residual $\|J_{\mathcal{G}}[\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}] - \mathcal{F}_t^{-1} \boldsymbol{G}_t\|_F^2$, whose normal equations $J_{\mathcal{G}}^* J_{\mathcal{G}} [\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}] = J_{\mathcal{G}}^* (\mathcal{F}_t^{-1} \boldsymbol{G}_t)$ coincide with (7) only when $\mathcal{F}_t = \boldsymbol{I}$ but generally differ for other $\mathcal{F}_t$. Its AdamW variant pairs a non-trivial $\mathcal{F}_t$ on $\boldsymbol{W}$ with the Frobenius (rather than $\mathcal{F}_t$-weighted) residual, mismatching the preconditioner's metric, and explicitly maintains $\boldsymbol{W}$-space first/second moments at $\mathcal{O}(mn)$ memory prohibitive at LLM scale. In contrast, our (11) measures the residual under the $\mathcal{F}_t$-induced $\mathcal{H}_t$-norm consistent with the preconditioner.

Manifold-based methods on $\mathcal{M}_r$. Rather than solving (7) on the factor space, this line of work performs Riemannian gradient descent on the rank-r matrix manifold $\mathcal{M}_r$. Riemannian Muon [5] uses retraction-based Muon updates on $\mathcal{M}_r$, applying Muon orthogonalization (replacing all singular values by 1) on the tangent space. The resulting step is equivalent to a per-step spectral $\boldsymbol{W}$-space preconditioner $\mathcal{F}_t = (\boldsymbol{G}_t \boldsymbol{G}_t^\top)^{1/2}$ (no accumulation across steps). RAdaGrad / RAdamW [4] run Riemannian gradient descent on $\mathcal{M}_r$ under a Shampoo $\boldsymbol{W}$-space preconditioner $\mathcal{F}_t = (\boldsymbol{L}_{\mathrm{Sh}} \otimes \boldsymbol{R}_{\mathrm{Sh}})^{\frac{1}{4}}$ restricted to the manifold tangent space, achieving a similar $\mathcal{F}_t$-aware behaviour to ours but via a retraction step on $\mathcal{M}_r$ instead of a closed-form solution of (7) in factor coordinates.

Other directions (LoRA-RITE / LoRA-GA). LoRA-RITE [37] introduces transformation invariance via a polar-decomposition-based reparameterization of the factor coordinates, with $\mathcal{F}_t = \boldsymbol{I}$; LoRA-GA [33] addresses initialization through spectral alignment with full fine-tuning gradients.

These methods reveal a recurring trade-off: cheap factor-space schemes either bypass (7) or take $\mathcal{F}_t = \boldsymbol{I}$ with block-diagonal approximations of $J_{\mathcal{G}}^* J_{\mathcal{G}}$, while methods admitting a non-trivial $\mathcal{F}_t$, such as LoRA-Pro AdamW, pay $\mathcal{O}(mn)$ memory or operate in the ambient $\boldsymbol{W}$-space. A gradient-statistics-aware $\mathcal{F}_t$ paired with $\mathcal{O}((m+n)r)$ memory in the LoRA factor space remains an underexplored design point, which our method (§ 3) targets via the Adafactor diagonal Kronecker $\mathcal{F}_t = \boldsymbol{L}_t^{1/2} \otimes \boldsymbol{R}_t^{1/2}$ together with a closed-form solution of (7) that picks a specific element of the r^2-dimensional affine solution set.

2.3 Choosing $\mathcal{F}_t$: Adaptive Preconditioner Toolkit on $\boldsymbol{W}$

The gap identified above asks for an $\mathcal{F}_t$ that is gradient-statistics-based yet cheap on $\boldsymbol{W}$. We review the standard families of $\boldsymbol{W}$-space preconditioners, organized by memory cost. All families construct a second-moment-based preconditioner $\mathcal{F}_t : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ from gradient outer-product statistics of the form $\boldsymbol{G}_t\boldsymbol{G}_t^\top$ or $\boldsymbol{G}_t \odot \boldsymbol{G}_t$, and produce the preconditioned update $\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta_t \mathcal{F}_t^{-1}\boldsymbol{G}_t$; they differ in the structure imposed on $\mathcal{F}_t$, which trades off expressiveness against cost. AdaGrad [8] and Adam [18] approximate $\mathcal{F}_t$ by its diagonal $\boldsymbol{h}_t \in \mathbb{R}^{mn}$ as an exponential moving average of $\boldsymbol{G}_t \odot \boldsymbol{G}_t$, yielding per-coordinate rescaling that ignores the matrix structure of $\boldsymbol{G}_t$ at memory $\mathcal{O}(mn)$. Adafactor [29] compresses this further into a rank-1 Kronecker form by maintaining only the row sums $\boldsymbol{l}_t \in \mathbb{R}^m$ and column sums $\boldsymbol{r}_t \in \mathbb{R}^n$ of $\boldsymbol{G}_t \odot \boldsymbol{G}_t$ (the elementwise Hadamard product), dropping the memory cost to $\mathcal{O}(m+n)$. Shampoo [11] maintains $\boldsymbol{L}_{\mathrm{Sh},t} = \boldsymbol{L}_{\mathrm{Sh},t-1} + \boldsymbol{G}_t\boldsymbol{G}_t^\top \in \mathbb{R}^{m\times m}$ and $\boldsymbol{R}_{\mathrm{Sh},t} = \boldsymbol{R}_{\mathrm{Sh},t-1} + \boldsymbol{G}_t^\top\boldsymbol{G}_t \in \mathbb{R}^{n\times n}$ and updates by $\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \boldsymbol{L}_{\mathrm{Sh},t}^{-\frac{1}{4}}\boldsymbol{G}_t\boldsymbol{R}_{\mathrm{Sh},t}^{-\frac{1}{4}}$; SOAP [23, 31] runs Adam in the eigenbasis of the Shampoo preconditioner; and K-FAC [9, 21] factorizes $\mathcal{F}_t$ as the Kronecker product of activation and gradient covariances. All three impose $\mathcal{O}(m^2+n^2)$ memory and $\mathcal{O}(m^3+n^3)$ per-step inverse cost, which dominates LoRA's budgets. Among these candidates, the Adafactor diagonal Kronecker form $\mathcal{F}_t = (\boldsymbol{L}_t \otimes \boldsymbol{R}_t)^{\frac{1}{2}}$ is the only one that is simultaneously gradient-statistics-based and cheap, requiring $\mathcal{O}(m+n)$ memory. Our method (§ 3) adopts this candidate and pairs it with a closed-form solution of the linear system (7) that respects the LoRA factorization.

3 FILO-Adafactor: An Instantiation of FILO

We instantiate the framework FILO (7) as **FILO-Adafactor** with two specific choices: (i) for the $\boldsymbol{W}$-space preconditioner, the Adafactor diagonal Kronecker form $\mathcal{H}_t = \boldsymbol{L}_t^{1/2} \otimes \boldsymbol{R}_t^{1/2}$ on $\boldsymbol{W}$ (§ 3.1); (ii) for the element of the affine solution set of (7), the unique minimizer of the $\mathcal{H}_t$-imbalance criterion (Solution 2). Choice (i) solves the linear system (7) through an equivalent weighted least-squares problem (Obstruction 1); choice (ii) resolves the r^2-dimensional ambiguity over $\ker(J_{\mathcal{G}})$ (Obstruction 2). The closed-form factor update is given in Theorem 3.2. Figure 1 contrasts the resulting $\boldsymbol{W}$-update geometry against LoRA-Pro and Riemannian Preconditioned LoRA [38] under the $\mathcal{H}_t$-weighted inner product.

3.1 The Adafactor Preconditioner $\mathcal{H}_t = \boldsymbol{L}_t^{1/2} \otimes \boldsymbol{R}_t^{1/2}$

To distinguish our choice from those of existing methods, we denote the preconditioner used in FILO-Adafactor by $\mathcal{H}_t$. Specifically, we adopt the diagonal Kronecker preconditioner $\mathcal{F}_t = \mathcal{H}_t = (\boldsymbol{L}_t \otimes \boldsymbol{R}_t)^{1/2}$ on $\boldsymbol{W}$, where $\boldsymbol{L}_t, \boldsymbol{R}_t$ are the Adafactor [29] rank-1 second-moment estimate of $\boldsymbol{G}_t \odot \boldsymbol{G}_t$:

$$\begin{gathered}\boldsymbol{L}_t = \operatorname{diag}(\boldsymbol{l}_t/\sqrt{\|\boldsymbol{l}_t\|_1}), \quad \boldsymbol{R}_t = \operatorname{diag}(\boldsymbol{r}_t/\sqrt{\|\boldsymbol{r}_t\|_1}), \quad \text{with} \\ \boldsymbol{l}_t = \beta_1\boldsymbol{l}_{t-1} + (1-\beta_1)\sum_{j=1}^{n}(\boldsymbol{G}_t \odot \boldsymbol{G}_t)_{i,j}, \quad \boldsymbol{r}_t = \beta_2\boldsymbol{r}_{t-1} + (1-\beta_2)\sum_{i=1}^{m}(\boldsymbol{G}_t \odot \boldsymbol{G}_t)_{i,j},\end{gathered} \tag{8}$$

where $\odot$ denotes the Hadamard product, $\|\cdot\|_1$ denotes the ℓ_1-norm, and $\beta_1, \beta_2 \in [0,1]$ are decay rates. The vectors $\boldsymbol{l}_t \in \mathbb{R}^m$ and $\boldsymbol{r}_t \in \mathbb{R}^n$ are the diagonals of the moving averages of $\boldsymbol{G}_t\boldsymbol{G}_t^\top$ and $\boldsymbol{G}_t^\top\boldsymbol{G}_t$, respectively, and together define the normalized rank-1 second-moment estimate $\boldsymbol{l}_t\boldsymbol{r}_t^\top/\sqrt{\|\boldsymbol{l}_t\|_1\|\boldsymbol{r}_t\|_1}$ following Adafactor [29]. The memory cost is $\mathcal{O}(m+n)$.

We treat $\mathcal{H}_t$ as an operator on $\mathbb{R}^{m\times n}$, defined by $\mathcal{H}_t\boldsymbol{Y} := \boldsymbol{L}_t^{1/2}\boldsymbol{Y}\boldsymbol{R}_t^{1/2}$ for any $\boldsymbol{Y} \in \mathbb{R}^{m\times n}$, with inverse $\mathcal{H}_t^{-1}\boldsymbol{K} = \boldsymbol{L}_t^{-1/2}\boldsymbol{K}\boldsymbol{R}_t^{-1/2}$. The $1/2$-power form yields the preconditioned direction $\mathcal{H}_t^{-1}\boldsymbol{G}_t = \boldsymbol{L}_t^{-1/2}\boldsymbol{G}_t\boldsymbol{R}_t^{-1/2}$, following Adafactor's inverse-square-root second-moment scaling [29]. This choice of exponent is also consistent with the $1/2$-power Shampoo variant studied in SOAP [31]. For full-matrix Shampoo, the Kronecker second-moment estimate corresponds, up to scaling, to one power-iteration step toward the Frobenius-optimal Kronecker approximation of the accumulated outer products of vectorized gradients [23]. We define the $\mathcal{H}_t$-weighted inner product by

$$\langle \boldsymbol{Y}, \boldsymbol{Z}\rangle_{\mathcal{H}_t} := \langle \mathcal{H}_t\boldsymbol{Y}, \boldsymbol{Z}\rangle = \langle \boldsymbol{L}_t^{1/2}\boldsymbol{Y}\boldsymbol{R}_t^{1/2}, \boldsymbol{Z}\rangle, \tag{9}$$

where $\langle\cdot,\cdot\rangle$ is the Frobenius inner product.

3.2 Solving the Linear System on Factor Space

With $\mathcal{H}_t = \boldsymbol{L}_t^{1/2} \otimes \boldsymbol{R}_t^{1/2}$ from § 3.1, the factor-space linear system (7) becomes

$$J_{\mathcal{G}}^* \mathcal{H}_t J_{\mathcal{G}} \left[\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}\right] \;=\; J_{\mathcal{G}}^*(\boldsymbol{G}_t), \tag{10}$$

in the candidate factor update $[\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}] \in \mathbb{R}^{m\times r} \times \mathbb{R}^{r\times n}$. The operator $J_{\mathcal{G}}^* \mathcal{H}_t J_{\mathcal{G}}$ is singular (Obstruction 1), so we cannot invert it.

Solution 1 (Resolve Obstruction 1: solve the equivalent least-squares problem). Equation (10) is the normal equation of

$$\min_{\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}} \left\| J_{\mathcal{G}}[\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}] - \mathcal{H}_t^{-1} \boldsymbol{G}_t \right\|_{\mathcal{H}_t}^2, \tag{11}$$

so solving (11) replaces inverting $J_{\mathcal{G}}^* \mathcal{H}_t J_{\mathcal{G}}$.

The following theorem characterizes the solution set of (11).

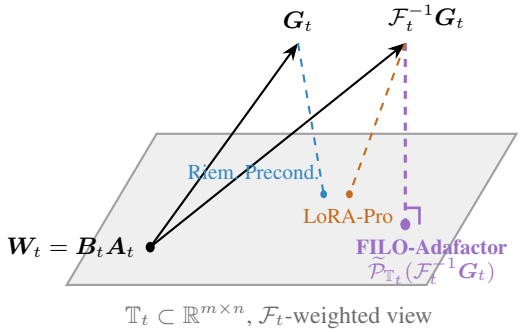


Figure 1: Geometric contrast of LoRA optimizers under the $\mathcal{F}_t$-weighted inner product on $\mathbb{R}^{m\times n}$; $\mathcal{F}_t^{-1}\boldsymbol{G}_t$ is the gradient under this inner product, and all updates land in $\mathbb{T}_t = \mathrm{range}(J_{\mathcal{G}})$. From $\mathcal{F}_t^{-1}\boldsymbol{G}_t$, **FILO-Adafactor** drops $\mathcal{F}_t$-orthogonally; **LoRA-Pro** does not realize an orthogonal projection under the $\mathcal{F}_t$-weighted inner product. **Riem. Precond.** lies in $\mathbb{T}_t$ but is neither a Frobenius nor an $\mathcal{F}_t$-weighted orthogonal projection. LoRA-Pro and FILO-Adafactor coincide when $\mathcal{F}_t = \boldsymbol{I}$.

Theorem 3.1 (Solution set of (11))**.** *Let* $\widetilde{\boldsymbol{G}}_t := \mathcal{H}_t^{-1}\boldsymbol{G}_t$*. Since* $J_{\mathcal{G}}[\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}] \in \mathbb{T}_t = \mathrm{range}(J_{\mathcal{G}})$*, the minimum of* (11) *is attained iff*

$$\boldsymbol{\Delta}_{\boldsymbol{B}_t}\boldsymbol{A}_t + \boldsymbol{B}_t\boldsymbol{\Delta}_{\boldsymbol{A}_t} \;=\; \widetilde{\mathcal{P}}_{\mathbb{T}_t}(\widetilde{\boldsymbol{G}}_t), \tag{12}$$

the $\mathcal{H}_t$*-orthogonal projection of* $\widetilde{\boldsymbol{G}}_t$ *onto* $\mathbb{T}_t$ *(closed form in Proposition C.5). The minimizers form an* r^2*-parameter family (Appendix C.5, Lemma C.1)*

$$\begin{aligned}
\boldsymbol{\Delta}_{\boldsymbol{B}_t}(\boldsymbol{X}_t) &= \left[\boldsymbol{L}_t^{-1/2} - \boldsymbol{B}_t(\boldsymbol{B}_t^\top \boldsymbol{L}_t^{1/2}\boldsymbol{B}_t)^{-1}\boldsymbol{B}_t^\top\right] \boldsymbol{G}_{\boldsymbol{B}_t}(\boldsymbol{A}_t\boldsymbol{R}_t^{1/2}\boldsymbol{A}_t^\top)^{-1} + \boldsymbol{B}_t\boldsymbol{X}_t, \\
\boldsymbol{\Delta}_{\boldsymbol{A}_t}(\boldsymbol{X}_t) &= (\boldsymbol{B}_t^\top \boldsymbol{L}_t^{1/2}\boldsymbol{B}_t)^{-1}\boldsymbol{G}_{\boldsymbol{A}_t}\boldsymbol{R}_t^{-1/2} - \boldsymbol{X}_t\boldsymbol{A}_t,
\end{aligned} \qquad \boldsymbol{X}_t \in \mathbb{R}^{r\times r}, \tag{13}$$

where the offsets $(\boldsymbol{B}_t\boldsymbol{X}_t, -\boldsymbol{X}_t\boldsymbol{A}_t)$ *parameterize* $\ker(J_{\mathcal{G}})$.

By Theorem 3.1, every factor pair in (13) induces the common $\boldsymbol{W}$-update $\widetilde{\mathcal{P}}_{\mathbb{T}_t}(\widetilde{\boldsymbol{G}}_t)$. The following solution selects a specific $\boldsymbol{X}_t$ to resolve this r^2-dimensional ambiguity (Obstruction 2).

Solution 2 ($\mathcal{H}_t$-balance fixes the $\ker(J_{\mathcal{G}})$ ambiguity). Among the $\boldsymbol{X}_t$-family (13), we select the unique element by choosing $\boldsymbol{X}_t$ to minimize the $\mathcal{H}_t$-imbalance $\|\boldsymbol{\Delta}_{\boldsymbol{B}_t}\boldsymbol{A}_t - \boldsymbol{B}_t\boldsymbol{\Delta}_{\boldsymbol{A}_t}\|_{\mathcal{H}_t}^2$ between the two factor contributions to the $\boldsymbol{W}$-update.

This criterion balances the magnitudes of the two factor contributions, in the same spirit as the regularizer in Imbalance-Regularized LoRA [41] and the standard balance term used in nonconvex low-rank matrix recovery [30]. Combining Theorem 3.1 with Solutions 1 and 2 fixes $\boldsymbol{X}_t$ in closed form and yields the full FILO-Adafactor update.

Theorem 3.2 (FILO-Adafactor closed-form factor update)**.** *The unique factor update solving* (11) *together with the* $\mathcal{H}_t$*-balance criterion (Solution 2) is*

$$\boldsymbol{\Delta}_{\boldsymbol{B}_t}^{\mathrm{opt}} = \left(\boldsymbol{I} - \tfrac{1}{2}\widetilde{\boldsymbol{P}}_{B_t}\right)\boldsymbol{L}_t^{-1/2}\boldsymbol{G}_{\boldsymbol{B}_t}(\boldsymbol{A}_t\boldsymbol{R}_t^{1/2}\boldsymbol{A}_t^\top)^{-1}, \quad \boldsymbol{\Delta}_{\boldsymbol{A}_t}^{\mathrm{opt}} = (\boldsymbol{B}_t^\top \boldsymbol{L}_t^{1/2}\boldsymbol{B}_t)^{-1}\boldsymbol{G}_{\boldsymbol{A}_t}\boldsymbol{R}_t^{-1/2}\left(\boldsymbol{I} - \tfrac{1}{2}\widetilde{\boldsymbol{Q}}_{A_t}\right),$$

Table 2: GPT-2 fine-tuning on E2E at $r = 4$, across model size (small / medium). Bold/underline = best/second-best per metric per (model, optimizer family). DART and cross-rank ablation ($r \in \{16, 64\}$) results are reported in Table 3 and Table 10.

Model	r	Method	E2E				
			BLEU	NIST	MET	ROUGE-L	CIDEr
GPT-2 small	4	SGD	54.8	4.56	34.0	63.3	1.29
		Scaled GD	<u>68.5</u>	<u>8.72</u>	<u>45.5</u>	69.4	2.40
		LoRA-Pro SGD	68.4	<u>8.72</u>	<u>45.5</u>	<u>69.6</u>	<u>2.43</u>
		FILO-Adafactor (ours)	**69.5**	**8.77**	**46.5**	**71.5**	**2.50**
		AdamW	69.1	8.75	46.0	70.5	2.47
		Scaled AdamW	<u>69.5</u>	<u>8.80</u>	<u>46.2</u>	<u>70.9</u>	<u>2.48</u>
		LoRA-Pro AdamW	69.2	8.73	45.9	70.8	2.47
		FILO-AdafactorM (ours)	**70.0**	**8.84**	**46.3**	**71.3**	**2.50**
GPT-2 medium	4	SGD	66.6	8.54	44.2	68.2	2.32
		Scaled GD	69.2	8.71	46.3	<u>70.9</u>	2.48
		LoRA-Pro SGD	<u>69.7</u>	<u>8.77</u>	<u>46.5</u>	<u>70.9</u>	<u>2.50</u>
		FILO-Adafactor (ours)	**70.3**	**8.84**	**46.9**	**71.7**	**2.54**
		AdamW	68.9	8.69	46.5	71.3	2.51
		Scaled AdamW	69.6	8.77	<u>46.6</u>	**71.8**	<u>2.52</u>
		LoRA-Pro AdamW	<u>69.8</u>	<u>8.78</u>	46.5	<u>71.7</u>	<u>2.52</u>
		FILO-AdafactorM (ours)	**70.3**	**8.84**	**46.7**	**71.8**	**2.53**

where the $\mathcal{H}_t$-weighted projector matrices are

$$\widetilde{\boldsymbol{P}}_{B_t} := \boldsymbol{B}_t(\boldsymbol{B}_t^\top \boldsymbol{L}_t^{1/2}\boldsymbol{B}_t)^{-1}\boldsymbol{B}_t^\top \boldsymbol{L}_t^{1/2}, \qquad \widetilde{\boldsymbol{Q}}_{A_t} := \boldsymbol{R}_t^{1/2}\boldsymbol{A}_t^\top(\boldsymbol{A}_t\boldsymbol{R}_t^{1/2}\boldsymbol{A}_t^\top)^{-1}\boldsymbol{A}_t. \tag{14}$$

Under the $\mathcal{H}_t$-balance criterion (Solution 2), two features are worth highlighting: (i) given $\mathcal{H}_t$, the update depends on the gradient only through the low-rank factor gradients $\boldsymbol{G}_{\boldsymbol{A}_t}, \boldsymbol{G}_{\boldsymbol{B}_t}$, keeping per-step optimizer state at $\mathcal{O}((m+n)r)$; (ii) the $\frac{1}{2}$ coefficients on the projectors are the signature of the $\mathcal{H}_t$-balance choice. The base algorithm, **FILO-Adafactor**, is summarized as Algorithm 1 in the appendix; its computational complexity is analyzed in Appendix E. For practical use we also provide **FILO-AdafactorM** (Algorithm 2), a momentum variant that maintains first-moment EMAs of the factor gradients. It also includes bias correction and decoupled weight decay. In both variants, the second-moment statistics are computed from the weight-space gradient captured by a backward hook.

4 Experimental Results

We evaluate FILO-Adafactor and FILO-AdafactorM against representatives of the design families identified in Table 1: LoRA trained with SGD / AdamW, Scaled GD / Scaled AdamW [38], LoRA-Pro SGD / AdamW [34], and SOAP [31]. The Scaled baselines are the SGD and AdamW implementations of Riemannian Preconditioned LoRA. Three axes test the trade-off identified in § 2.2: model scale, from 124M–355M GPT-2 to 7B Mistral/Qwen2; task family, covering NLU, reasoning, math, generation, and image tasks; and resource cost, measured by peak GPU memory and per-step time. Following [34, 38], learning rates are independently tuned per optimizer via grid search; full hyperparameters are in Appendix F.1. Unless otherwise noted, all runs use PyTorch [26] on NVIDIA A100 GPUs.

4.1 Controlled study: GPT-2

We start with controlled fine-tuning of GPT-2 [27] (small, 124M; medium, 355M) on the E2E natural language generation challenge [25], sweeping rank $r \in \{4, 16, 64\}$ to evaluate the effectiveness of our preconditioning approach at small scale.

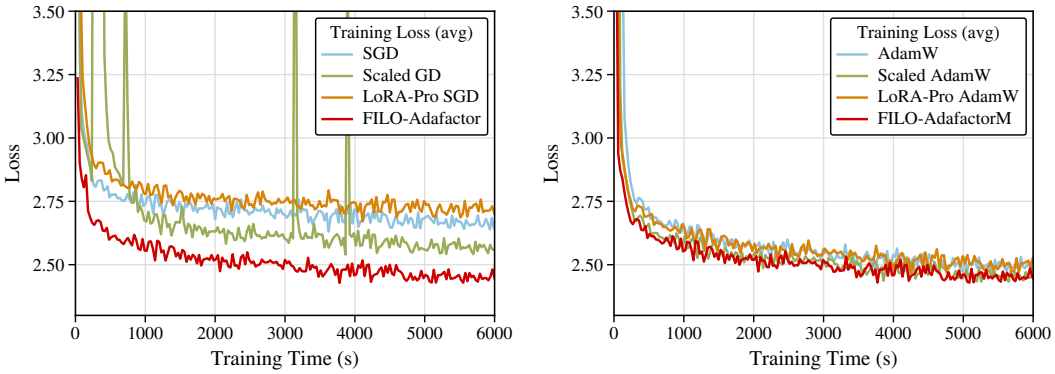
Table 2 reports E2E scores at $r = 4$ on both GPT-2 small and medium. FILO-Adafactor and FILO-AdafactorM achieve the best or tied-best score on every metric within their respective optimizer groups for both model sizes. To further validate the effectiveness of both variants, we also conducted GPT-2 fine-tuning experiments on DART [24]. Both variants achieve the highest BLEU scores within

Table 3: Scores of GPT-2 small model (rank=4) fine-tuned using different optimizers. Evaluation is conducted on DART dataset.

Methods	BLEU↑	METEOR↑	chrF++↑	TER↓	BLEURT↑
SGD	41.2	0.63	0.59	0.52	0.33
Scaled GD	43.8	**0.66**	0.61	0.50	0.38
LoRA-Pro SGD	44.1	**0.66**	0.61	0.50	0.38
FILO-Adafactor (ours)	**44.6**	**0.66**	**0.62**	**0.49**	**0.39**
AdamW	43.9	0.66	0.60	0.50	0.38
Scaled AdamW	44.8	**0.67**	**0.62**	0.49	**0.40**
LoRA-Pro AdamW	44.9	0.66	**0.62**	0.50	0.39
FILO-AdafactorM (ours)	**45.4**	**0.67**	0.60	**0.49**	**0.40**

their respective groups, as shown in Table 3. These results show that the gains extend to another generation benchmark. The cross-rank results also show that both variants retain the highest BLEU scores within their respective groups at $r \in \{16, 64\}$ on E2E, as reported in Table 10 in Appendix F.2.

Figure 2 compares training loss against cumulative training time on GPT-2 medium / E2E at $r = 4$. At a budget of 6000 s, FILO-Adafactor and FILO-AdafactorM attain lower training loss than the baselines in their respective panels.



(a) SGD-based methods.

(b) AdamW-based methods.

Figure 2: Training loss versus cumulative training time for GPT-2 medium on E2E ($r = 4$). We divide consecutive optimizer updates into non-overlapping blocks of 100. Each point shows a block's mean training loss at the cumulative training time of its final update. For each method, we use the learning rate with the lowest validation loss among all the tested learning rates. These experiments were run on H800 GPUs.

4.2 Extension to 7B-scale LLMs: Mistral-7B and Qwen2-7B

We next test whether the controlled-setting gains carry over to the 7B parameter scale, using Mistral-7B [16] and Qwen2-7B [36]. For Mistral-7B we fine-tune on the GLUE [32] tasks RTE, CoLA, and MRPC, with three random seeds on RTE. For Qwen2-7B we additionally evaluate on the reasoning benchmark ARC [6] and the math benchmark GSM8K [7].

Table 4 shows that FILO-AdafactorM achieves the best accuracy on six of the seven settings and ranks second on Mistral-7B CoLA, 0.1 percentage point behind Scaled AdamW. Notably, LoRA-Pro AdamW incurs $\mathcal{O}(mn)$ memory to store the full-weight gradient and its first and second moments, yet it trails AdamW on multiple Qwen2-7B tasks. To compare with an optimizer based on Kronecker preconditioning, we also include SOAP [31] in the 7B experiments. FILO-AdafactorM achieves higher accuracy than SOAP in all seven settings. Table 5 contrasts per-step GPU time and peak memory on Mistral-7B GLUE-RTE: FILO-AdafactorM matches Scaled AdamW's peak memory of 26.0 GB, while LoRA-Pro AdamW requires 50.4 GB, nearly twice as much.

Table 4: Mistral-7B and Qwen2-7B fine-tuning accuracy (%) with rank $r = 8$. Bold/underline = best/second-best.

	Mistral-7B			Qwen2-7B			
Method	RTE	CoLA	MRPC	RTE	MRPC	ARC	GSM8K
AdamW	<u>89.4</u>	69.4	89.5	<u>90.6</u>	<u>90.0</u>	84.3	75.1
Scaled AdamW	89.1	**71.5**	<u>89.7</u>	<u>90.6</u>	87.0	<u>85.3</u>	74.2
LoRA-Pro AdamW	88.8	68.5	84.8	88.1	89.2	80.9	<u>75.7</u>
SOAP	88.9	68.3	86.3	86.6	89.2	80.9	73.2
FILO-AdafactorM (ours)	**89.5**	<u>71.4</u>	**90.0**	**91.0**	**90.4**	**85.6**	**76.4**

Table 5: Per-step GPU time and peak GPU memory on Mistral-7B. FILO-Adafactor matches the LoRA-level memory footprint of vanilla LoRA (AdamW), while LoRA-Pro AdamW pays roughly $2\times$ peak memory for its full-weight gradient and moments.

Method	s/step	Mem (GB)
AdamW	0.24	25.8
Scaled AdamW	0.36	26.0
SOAP	0.65	26.4
LoRA-Pro AdamW	1.35	50.4
FILO-Adafactor	0.46	21.5
FILO-AdafactorM	0.99	26.0

Table 6: CLIP and FID scores on Mix-of-Show diffusion fine-tuning. Bold/underline = best/second-best.

Method	scaling=0.7		scaling=1	
	CLIP↑	FID↓	CLIP↑	FID↓
SGD	27.79	69.90	<u>31.40</u>	40.95
Scaled GD	<u>31.23</u>	35.86	30.60	29.62
LoRA-Pro SGD	**31.47**	<u>34.30</u>	30.48	<u>29.19</u>
FILO-Adafactor	**31.47**	**30.17**	**31.58**	**28.18**
AdamW	**31.47**	34.15	<u>30.68</u>	<u>27.80</u>
Scaled AdamW	24.21	48.23	24.51	34.18
LoRA-Pro AdamW	<u>31.04</u>	<u>29.18</u>	30.60	28.18
FILO-AdafactorM	**31.47**	**29.01**	**30.73**	**27.13**

4.3 Diffusion model personalization (Mix-of-Show)

To test transfer beyond NLP, we evaluate both variants of FILO-Adafactor on diffusion-model personalization with the Mix-of-Show framework [10], which uses Embedding Decomposed LoRA (EDLoRA) on the text encoder and U-Net of a Stable Diffusion backbone. Setup follows [10, 38] with embedding tuning disabled. We report CLIP score [13] and FID [14] at LoRA scaling factors 0.7 and 1.0. Higher CLIP scores indicate better alignment with the prompt, while lower FID scores indicate greater distributional similarity to reference images.

Table 6 shows that both variants of FILO-Adafactor achieve the lowest FID at every scaling-optimizer combination and the best CLIP at scaling 1.0 in both SGD- and AdamW-based families, while remaining competitive with the best baseline at scaling 0.7. This confirms that the gains from gradient-statistics-aware preconditioning carry over from text generation to image generation. Qualitative samples featuring Harry Potter and Hermione Granger are in Appendix G, along with per-prompt grids across LoRA scaling factors.

5 Conclusion

We introduced **FILO**, organizing existing LoRA optimizers along two axes: (i) the choice of $\boldsymbol{W}$-space preconditioner $\mathcal{F}_t$ (§ 2.2, Table 1), and (ii) the factor-space inversion strategy for the singular operator $J_{\mathcal{G}}^{*}\mathcal{F}_t J_{\mathcal{G}}$. This framework exposes a previously underexplored design point: pairing a non-trivial gradient-statistics-aware $\mathcal{F}_t$ with LoRA's $\mathcal{O}((m+n)r)$ optimizer-state budget. We instantiate FILO as **FILO-Adafactor**, combining an Adafactor diagonal Kronecker preconditioner $\mathcal{H}_t$ on $\boldsymbol{W}$ with an $\mathcal{H}_t$-balance criterion that selects a unique factor update from the affine solution set of (7). For the instantaneous-gradient solve, the induced first-order weight update is the closest LoRA approximation to the preconditioned $\boldsymbol{W}$-space direction under the $\mathcal{H}_t$-weighted norm, and admits a closed-form expression. Across GPT-2 on E2E, Mistral-7B and Qwen2-7B on GLUE, ARC, and GSM8K, and Mix-of-Show diffusion personalization, FILO-Adafactor and FILO-AdafactorM are competitive with or improve over a representative set of LoRA optimizers while keeping peak GPU memory at the LoRA optimizer level.

Acknowledgment

The work is partially supported by Hong Kong Research Grant Council GRF grants 16307023, 16306124, and 16307325.

References

[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. Optimization algorithms on matrix manifolds. In *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2009.

[2] 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.

[3] Fengmiao Bian, Jian-Feng Cai, and Rui Zhang. A preconditioned riemannian gradient descent algorithm for low-rank matrix recovery. *SIAM Journal on Matrix Analysis and Applications*, 45 (4):2075–2103, 2024.

[4] Fengmiao Bian, Jinyang ZHENG, Ziyun Liu, Jianzhou Luo, and Jian-Feng CAI. Finding low-rank matrix weights in dnns via riemannian optimization: Radagrad and radamw. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, Main Conference, pages 59916–59950. Curran Associates, Inc., 2025. doi: 10.52202/085713-2004. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/5679173c400b332796426e443ab5ea0d-Paper-Conference.pdf.

[5] Vladimir Bogachev, Vladimir Aletov, Alexander Molozhavenko, Denis Bobkov, Vera Soboleva, Aibek Alanov, and Maxim Rakhuba. Lora meets riemannion: Muon optimizer for parametrization-independent low-rank adapters. *arXiv preprint arXiv:2507.12142*, 2025.

[6] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

[7] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

[8] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2011.

[9] Runa Eschenhagen, Alexander Immer, Richard Turner, Frank Schneider, and Philipp Hennig. Kronecker-factored approximate curvature for modern neural network architectures. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 33624–33655, 2023.

[10] Yuchao Gu, Xintao Wang, Jay Zhangjie Wu, Yujun Shi, Yunpeng Chen, Zihan Fan, Wuyou Xiao, Rui Zhao, Shuning Chang, Weijia Wu, et al. Mix-of-show: Decentralized low-rank adaptation for multi-concept customization of diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 15890–15902, 2023.

[11] Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning (ICML)*, pages 1842–1850. PMLR, 2018.

[12] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models. In *International Conference on Machine Learning (ICML)*, pages 17783–17806. PMLR, 2024.

[13] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP (1)*, 2021.

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems (NeurIPS)*, volume 30, 2017.

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, page 3, 2022.

[16] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL `https://arxiv.org/abs/2310.06825`.

[17] Hiroyuki Kasai, Pratik Jawanpuria, and Bamdev Mishra. Riemannian adaptive stochastic gradient algorithms on matrix manifolds. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3262–3271. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/kasai19a.html`.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Frederik Kunstner, Philipp Hennig, and Lukas Balles. Limitations of the empirical fisher approximation for natural gradient descent. *Advances in neural information processing systems*, 32, 2019.

[20] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[21] James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning (ICML)*, pages 2408–2417. PMLR, 2015.

[22] Zhanfeng Mo, Long-Kai Huang, and Sinno Jialin Pan. Parameter and memory efficient pretraining via low-rank riemannian optimization. In *International Conference on Learning Representations (ICLR)*, 2025.

[23] Depen Morwani, Itai Shapira, Nikhil Vyas, Eran Malach, Sham Kakade, and Lucas Janson. A new perspective on shampoo's preconditioner. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations (ICLR)*, volume 2025, pages 5802–5822, 2025. URL `https://proceedings.iclr.cc/paper_files/paper/2025/file/11332b6b6cf4485b84afadb1352d3a9a-Paper-Conference.pdf`.

[24] Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, et al. Dart: Open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 432–447, 2021.

[25] Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. The e2e dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*, 2017.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems (NeurIPS)*, volume 32, 2019.

[27] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning (ICML)*, pages 8748–8763. PMLR, 2021.

[29] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning (ICML)*, pages 4596–4604. PMLR, 2018.

[30] Stephen Tu, Ross Boczar, Max Simchowitz, Mahdi Soltanolkotabi, and Ben Recht. Low-rank solutions of linear matrix equations via procrustes flow. In *International conference on machine learning*, pages 964–973. PMLR, 2016.

[31] Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham M Kakade. Soap: Improving and stabilizing shampoo using adam for language modeling. In *International Conference on Learning Representations (ICLR)*, 2025.

[32] 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, 2019. URL `https://arxiv.org/abs/1804.07461`.

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

[34] Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. Lora-pro: Are low-rank adapters properly optimized? In *International Conference on Learning Representations(ICLR)*, 2025.

[35] Ke Wei, Jian-Feng Cai, Tony F Chan, and Shingyu Leung. Guarantees of riemannian optimization for low rank matrix recovery. *SIAM Journal on Matrix Analysis and Applications*, 37(3): 1198–1222, 2016.

[36] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[37] 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 *The Thirteenth International Conference on Learning Representations*, 2025.

[38] Fangzhao Zhang and Mert Pilanci. Riemannian preconditioned lora for fine-tuning foundation models. In *International Conference on Machine Learning (ICML)*, 2024.

[39] Yuanhe Zhang, Fanghui Liu, and Yudong Chen. Lora-one: One-step full gradient could suffice for fine-tuning large language models, provably and efficiently. In *International Conference on Machine Learning (ICML)*, 2025.

[40] Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection. In *International Conference on Machine Learning (ICML)*, pages 61121–61143. PMLR, 2024.

[41] Zhenyu Zhu, Yongtao Wu, Quanquan Gu, and Volkan Cevher. Imbalance-regularized lora: A plug-and-play method for improving fine-tuning of foundation models. In *Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning*, 2024.

Contents

A Limitations and Broader Impact **14**

B Notation **14**

C Proof of Theoretical Results **14**

C.1 Computation of Jacobian 14

C.2 Proof of Proposition C.2 (Kernel of $J_{\mathcal{G}}$) 16

C.3 Optimality of $\mathcal{H}_t$-Projection (Solution 1) 17

C.4 Orthogonal Projection to Tangent Space 17

C.5 Proof of Theorem 3.2 (closed-form factor update) 19

D Algorithm **20**

E Computational and Memory Complexity Analysis of FILO-Adafactor **20**

F Supplementary Experiments of GPT-2 Fine-tuning **22**

F.1 Parameter Settings 22

F.2 Cross-rank ablation on GPT-2 small 22

F.3 Training Efficiency Comparison 23

G Supplementary Experiments of Diffusion Model Fine-tuning **23**

A Limitations and Broader Impact

Limitations. The rank-r constraint in low-rank adaptation methods inherently limits their expressiveness compared to full-parameter fine-tuning.

Broader Impact. The proposed methods demonstrate broad compatibility across diverse DNN architectures, which is useful to fields like healthcare, farming, and schools, especially in places with limited resources.

B Notation

Table 7 summarizes the notation used throughout the paper. Calligraphic letters denote linear operators on $\mathbb{R}^{m\times n}$ (e.g., $\mathcal{F}_t, \mathcal{H}_t$), while bold letters denote matrices (e.g., $\boldsymbol{B}_t, \boldsymbol{A}_t, \boldsymbol{L}_t, \boldsymbol{R}_t, \boldsymbol{G}_t$).

C Proof of Theoretical Results

C.1 Computation of Jacobian

Proposition C.1 (Computation of $J_{\mathcal{G}}$ and $J_{\mathcal{G}}^*$)**.** *Let $[\boldsymbol{B}, \boldsymbol{A}]$ be a pair of low-rank factors with $\boldsymbol{B} \in \mathbb{R}^{m\times r}, \boldsymbol{A} \in \mathbb{R}^{r\times n}$. Define the generator $\mathcal{G} : [\mathbb{R}^{m\times r}, \mathbb{R}^{r\times n}] \to \mathbb{R}^{m\times n}$ by $\mathcal{G}([\boldsymbol{B}, \boldsymbol{A}]) = \boldsymbol{B}\boldsymbol{A}$. Denote the Jacobian of $\mathcal{G}$ by $J_{\mathcal{G}}$ and its adjoint by $J_{\mathcal{G}}^*$. Then, for any $[\boldsymbol{P}, \boldsymbol{Q}] \in [\mathbb{R}^{m\times r}, \mathbb{R}^{r\times n}]$ and any $\boldsymbol{C} \in \mathbb{R}^{m\times n}$,*

- $J_{\mathcal{G}}([\boldsymbol{B}, \boldsymbol{A}])[\boldsymbol{P}, \boldsymbol{Q}] = \boldsymbol{P}\boldsymbol{A} + \boldsymbol{B}\boldsymbol{Q}$,
- $J_{\mathcal{G}}^*([\boldsymbol{B}, \boldsymbol{A}])(\boldsymbol{C}) = [\boldsymbol{C}\boldsymbol{A}^\top, \boldsymbol{B}^\top\boldsymbol{C}]$,
- $J_{\mathcal{G}}([\boldsymbol{B}, \boldsymbol{A}])J_{\mathcal{G}}^*([\boldsymbol{B}, \boldsymbol{A}])(\boldsymbol{C}) = \boldsymbol{C}\boldsymbol{A}^\top\boldsymbol{A} + \boldsymbol{B}\boldsymbol{B}^\top\boldsymbol{C}$.
- $J_{\mathcal{G}}^*([\boldsymbol{B}, \boldsymbol{A}])J_{\mathcal{G}}([\boldsymbol{B}, \boldsymbol{A}])[\boldsymbol{P}, \boldsymbol{Q}] = [\boldsymbol{P}\boldsymbol{A}\boldsymbol{A}^\top + \boldsymbol{B}\boldsymbol{Q}\boldsymbol{A}^\top,\ \boldsymbol{B}^\top\boldsymbol{P}\boldsymbol{A} + \boldsymbol{B}^\top\boldsymbol{B}\boldsymbol{Q}]$.

Proof. The Jacobian operator $J_{\mathcal{G}}([\boldsymbol{B}, \boldsymbol{A}])[\boldsymbol{P}, \boldsymbol{Q}] : [\mathbb{R}^{m\times r}, \mathbb{R}^{r\times n}] \to \mathbb{R}^{m\times n}$ represents the derivative of $\mathcal{G}$ at $[\boldsymbol{B}, \boldsymbol{A}]$ along the direction $[\boldsymbol{P}, \boldsymbol{Q}]$. Similarly, $J_{\mathcal{G}}^*([\boldsymbol{B}, \boldsymbol{A}])(\boldsymbol{C}) : \mathbb{R}^{m\times n} \to [\mathbb{R}^{m\times r}, \mathbb{R}^{r\times n}]$ is the adjoint of $J_{\mathcal{G}}$ at $[\boldsymbol{B}, \boldsymbol{A}]$ along the direction $\boldsymbol{C}$. For more details, see [1, § 6.1].

Table 7: Notation used throughout the paper.

Symbol	Meaning
Dimensions and indices	
m, n	Output and input dimensions of the adapted weight $\boldsymbol{W} \in \mathbb{R}^{m\times n}$
r	LoRA rank, $r \ll \min(m, n)$
t	Optimization step index
Matrices (in bold)	
$\boldsymbol{W}_0, \boldsymbol{W}_t$	Frozen pretrained weight; LoRA increment at step t, $\boldsymbol{W}_t = \boldsymbol{B}_t\boldsymbol{A}_t$
$\boldsymbol{B}_t, \boldsymbol{A}_t$	LoRA factors, $\boldsymbol{B}_t \in \mathbb{R}^{m\times r}$, $\boldsymbol{A}_t \in \mathbb{R}^{r\times n}$
$\boldsymbol{G}_t$	Stochastic gradient $\nabla_{\boldsymbol{W}}\mathcal{L}(\boldsymbol{W}_0 + \boldsymbol{W}_t)$ on $\boldsymbol{W}$
$\boldsymbol{G}_{\boldsymbol{B}_t}, \boldsymbol{G}_{\boldsymbol{A}_t}$	Factor gradients $\boldsymbol{G}_t\boldsymbol{A}_t^\top$ and $\boldsymbol{B}_t^\top\boldsymbol{G}_t$
$\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}$	Factor updates: $\boldsymbol{B}_{t+1} = \boldsymbol{B}_t - \eta_t\boldsymbol{\Delta}_{\boldsymbol{B}_t}$, similarly for $\boldsymbol{A}$
$\boldsymbol{\Delta}_t$	Induced $\boldsymbol{W}$-update $\boldsymbol{\Delta}_{\boldsymbol{B}_t}\boldsymbol{A}_t + \boldsymbol{B}_t\boldsymbol{\Delta}_{\boldsymbol{A}_t}$
$\boldsymbol{l}_t, \boldsymbol{r}_t$	Adafactor row/column statistics, $\boldsymbol{l}_t \in \mathbb{R}^m$, $\boldsymbol{r}_t \in \mathbb{R}^n$
$\boldsymbol{L}_t, \boldsymbol{R}_t$	Adafactor diagonal preconditioner factors (Eq. (8))
$\boldsymbol{L}_{\mathrm{Sh},t}, \boldsymbol{R}_{\mathrm{Sh},t}$	Shampoo full-covariance factors $\sum_s \boldsymbol{G}_s\boldsymbol{G}_s^\top$, $\sum_s \boldsymbol{G}_s^\top\boldsymbol{G}_s$
$\boldsymbol{X}_t$	Free $r \times r$ matrix parameterizing the $\ker(J_{\mathcal{G}})$ orbit (Lemma C.1)
$\widetilde{\boldsymbol{P}}_{B_t}, \widetilde{\boldsymbol{Q}}_{A_t}$	Auxiliary $\boldsymbol{L}_t^{1/2}/\boldsymbol{R}_t^{1/2}$-weighted projector matrices in Theorem 3.2
Operators (in calligraphic)	
$\mathcal{F}_t$	Preconditioner on $\boldsymbol{W}$, $\mathcal{F}_t : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$
$\mathcal{H}_t$	Preconditioner used in FILO-Adafactor, $\mathcal{H}_t\boldsymbol{Y} = \boldsymbol{L}_t^{1/2}\boldsymbol{Y}\boldsymbol{R}_t^{1/2}$
$\mathcal{E}_t$	Empirical Fisher on the factor space $[\boldsymbol{B}_t, \boldsymbol{A}_t]$, related to $\mathcal{F}_t$ via $\mathcal{E}_t = J_{\mathcal{G}}^*\mathcal{F}_t J_{\mathcal{G}}$ (Eq. (4))
$\widetilde{\mathcal{P}}_{\mathbb{T}_t}, \widetilde{\mathcal{P}}_{\mathbb{T}_t}^{\perp}$	$\mathcal{H}_t$-orthogonal projector onto $\mathbb{T}_t$ and its complement
$\mathcal{P}_{\mathbb{T}_t}$	Frobenius (Euclidean) orthogonal projector onto $\mathbb{T}_t$
$J_{\mathcal{G}}, J_{\mathcal{G}}^*, J_{\mathcal{G}}^\dagger$	Jacobian of $\mathcal{G} : [\boldsymbol{B}, \boldsymbol{A}] \mapsto \boldsymbol{B}\boldsymbol{A}$, its adjoint, and Moore–Penrose pseudoinverse
Subspaces and inner products	
$\mathbb{T}_t$	$\mathrm{Im}(J_{\mathcal{G}}) = \{\boldsymbol{P}\boldsymbol{A}_t + \boldsymbol{B}_t\boldsymbol{Q}\} \subset \mathbb{R}^{m\times n}$, range of one LoRA step
$\mathcal{M}_r$	Manifold of rank-r matrices in $\mathbb{R}^{m\times n}$
$\langle\cdot,\cdot\rangle$	Frobenius inner product on $\mathbb{R}^{m\times n}$
$\langle\cdot,\cdot\rangle_{\mathcal{H}_t}, \|\cdot\|_{\mathcal{H}_t}$	$\mathcal{H}_t$-weighted inner product and norm (Eq. (9))
$\widetilde{\boldsymbol{G}}_t$	Adafactor-preconditioned direction $\mathcal{H}_t^{-1}\boldsymbol{G}_t = \boldsymbol{L}_t^{-1/2}\boldsymbol{G}_t\boldsymbol{R}_t^{-1/2}$
Other	
η_t	Learning rate at step t
β_1, β_2	EMA decay rates for Adafactor statistics $\boldsymbol{l}_t, \boldsymbol{r}_t$
$\odot$	Hadamard (elementwise) product
$\mathrm{diag}(\boldsymbol{v})$	Diagonal matrix with $\boldsymbol{v}$ on the diagonal
$\mathcal{L}$	Loss function

(i) The computation of $J_{\mathcal{G}}$. Let $\boldsymbol{B}(t) : \mathbb{R} \to \mathbb{R}^{m\times r}$ and $\boldsymbol{A}(t) : \mathbb{R} \to \mathbb{R}^{r\times n}$ be differentiable curves with $\boldsymbol{B}(0) = \boldsymbol{B}$ and $\boldsymbol{A}(0) = \boldsymbol{A}$. By the chain rule, the Jacobian of $\mathcal{G}$ at $[\boldsymbol{B}, \boldsymbol{A}]$ along these curves is

$$\begin{aligned} J_{\mathcal{G}}([\boldsymbol{B}(t), \boldsymbol{A}(t)])[\dot{\boldsymbol{B}}(t), \dot{\boldsymbol{A}}(t)]\Big|_{t=0} &= \left[\frac{\mathrm{d}\mathcal{G}([\boldsymbol{B}, \boldsymbol{A}])}{\mathrm{d}\boldsymbol{B}}\right]\dot{\boldsymbol{B}}(t)\Big|_{t=0} + \left[\frac{\mathrm{d}\mathcal{G}([\boldsymbol{B}, \boldsymbol{A}])}{\mathrm{d}\boldsymbol{A}}\right]\dot{\boldsymbol{A}}(t)\Big|_{t=0} \\ &= \dot{\boldsymbol{B}}(t)\boldsymbol{A}(t)\Big|_{t=0} + \boldsymbol{B}(t)\dot{\boldsymbol{A}}(t)\Big|_{t=0} \\ &= \dot{\boldsymbol{B}}(0)\boldsymbol{A} + \boldsymbol{B}\dot{\boldsymbol{A}}(0), \end{aligned}$$

where $\dot{\boldsymbol{B}}(t)$ and $\dot{\boldsymbol{A}}(t)$ denote the derivatives of $\boldsymbol{B}(t)$ and $\boldsymbol{A}(t)$ with respect to t. The second line follows because $\mathcal{G}([\boldsymbol{B}, \boldsymbol{A}]) = \boldsymbol{B}\boldsymbol{A}$, hence $\frac{\mathrm{d}\mathcal{G}([\boldsymbol{B},\boldsymbol{A}])}{\mathrm{d}\boldsymbol{B}}$ and $\frac{\mathrm{d}\mathcal{G}([\boldsymbol{B},\boldsymbol{A}])}{\mathrm{d}\boldsymbol{A}}$ are both linear operators.

Since $\dot{\boldsymbol{B}}(0)$ and $\dot{\boldsymbol{A}}(0)$ are arbitrary, for any $[\boldsymbol{P}, \boldsymbol{Q}] \in [\mathbb{R}^{m\times r}, \mathbb{R}^{r\times n}]$, we obtain

$$J_{\mathcal{G}}([\boldsymbol{B}, \boldsymbol{A}])[\boldsymbol{P}, \boldsymbol{Q}] = \boldsymbol{P}\boldsymbol{A} + \boldsymbol{B}\boldsymbol{Q}.$$

(ii) The computation of $J_{\mathcal{G}}^*$. For brevity, write $J_{\mathcal{G}}[\boldsymbol{P}, \boldsymbol{Q}]$ for $J_{\mathcal{G}}([\boldsymbol{B}, \boldsymbol{A}])[\boldsymbol{P}, \boldsymbol{Q}]$ and $J_{\mathcal{G}}^*(\boldsymbol{C})$ for $J_{\mathcal{G}}^*([\boldsymbol{B}, \boldsymbol{A}])(\boldsymbol{C})$. By definition of the adjoint (with respect to the Frobenius inner product),

for any $[\boldsymbol{P}, \boldsymbol{Q}] \in (\mathbb{R}^{m\times r}, \mathbb{R}^{r\times n})$ and $\boldsymbol{C} \in \mathbb{R}^{m\times n}$,

$$\big\langle J_{\mathcal{G}}[\boldsymbol{P}, \boldsymbol{Q}], \boldsymbol{C}\big\rangle = \big\langle [\boldsymbol{P}, \boldsymbol{Q}], J^*_{\mathcal{G}}(\boldsymbol{C})\big\rangle.$$

For the left-hand side,

$$\begin{aligned}\big\langle J_{\mathcal{G}}[\boldsymbol{P}, \boldsymbol{Q}], \boldsymbol{C}\big\rangle &= \big\langle \boldsymbol{P}\boldsymbol{A} + \boldsymbol{B}\boldsymbol{Q}, \boldsymbol{C}\big\rangle \\ &= \big\langle \boldsymbol{P}\boldsymbol{A}, \boldsymbol{C}\big\rangle + \big\langle \boldsymbol{B}\boldsymbol{Q}, \boldsymbol{C}\big\rangle \\ &= \big\langle \boldsymbol{P}, \boldsymbol{C}\boldsymbol{A}^\top\big\rangle + \big\langle \boldsymbol{Q}, \boldsymbol{B}^\top\boldsymbol{C}\big\rangle.\end{aligned}$$

For the right-hand side, writing $J^*_{\mathcal{G}}(\boldsymbol{C}) = [\boldsymbol{C}_1, \boldsymbol{C}_2]$, then

$$\begin{aligned}\big\langle [\boldsymbol{P}, \boldsymbol{Q}], J^*_{\mathcal{G}}(\boldsymbol{C})\big\rangle &= \big\langle [\boldsymbol{P}, \boldsymbol{Q}], [\boldsymbol{C}_1, \boldsymbol{C}_2]\big\rangle \\ &= \big\langle \boldsymbol{P}, \boldsymbol{C}_1\big\rangle + \big\langle \boldsymbol{Q}, \boldsymbol{C}_2\big\rangle.\end{aligned}$$

Hence $\boldsymbol{C}_1 = \boldsymbol{C}\boldsymbol{A}^\top$ and $\boldsymbol{C}_2 = \boldsymbol{B}^\top\boldsymbol{C}$, and therefore $J^*_{\mathcal{G}}([\boldsymbol{B}, \boldsymbol{A}])(\boldsymbol{C}) = [\boldsymbol{C}\boldsymbol{A}^\top, \boldsymbol{B}^\top\boldsymbol{C}]$.

(iii) By (i)–(ii), $J_{\mathcal{G}}([\boldsymbol{B}, \boldsymbol{A}])J^*_{\mathcal{G}}([\boldsymbol{B}, \boldsymbol{A}])(\boldsymbol{C}) = J_{\mathcal{G}}([\boldsymbol{B}, \boldsymbol{A}])[\boldsymbol{C}\boldsymbol{A}^\top, \boldsymbol{B}^\top\boldsymbol{C}] = \boldsymbol{C}\boldsymbol{A}^\top\boldsymbol{A} + \boldsymbol{B}\boldsymbol{B}^\top\boldsymbol{C}$ as claimed.

(iv) By (i)–(ii), $J^*_{\mathcal{G}}([\boldsymbol{B}, \boldsymbol{A}])J_{\mathcal{G}}([\boldsymbol{B}, \boldsymbol{A}])[\boldsymbol{P}, \boldsymbol{Q}] = J^*_{\mathcal{G}}([\boldsymbol{B}, \boldsymbol{A}])(\boldsymbol{P}\boldsymbol{A} + \boldsymbol{B}\boldsymbol{Q}) = [(\boldsymbol{P}\boldsymbol{A} + \boldsymbol{B}\boldsymbol{Q})\boldsymbol{A}^\top,\ \boldsymbol{B}^\top(\boldsymbol{P}\boldsymbol{A} + \boldsymbol{B}\boldsymbol{Q})] = [\boldsymbol{P}\boldsymbol{A}\boldsymbol{A}^\top + \boldsymbol{B}\boldsymbol{Q}\boldsymbol{A}^\top,\ \boldsymbol{B}^\top\boldsymbol{P}\boldsymbol{A} + \boldsymbol{B}^\top\boldsymbol{B}\boldsymbol{Q}]$ as claimed.

□

C.2 Proof of Proposition C.2 (Kernel of $J_{\mathcal{G}}$)

Proposition C.2 (Kernel of $J_{\mathcal{G}}$ from factorization redundancy)**.** *For $\boldsymbol{B}_t \in \mathbb{R}^{m\times r}$ of column rank r and $\boldsymbol{A}_t \in \mathbb{R}^{r\times n}$ of row rank r, the kernel of the Jacobian operator $J_{\mathcal{G}} = J_{\mathcal{G}}([\boldsymbol{B}_t, \boldsymbol{A}_t])$ is*

$$\ker(J_{\mathcal{G}}) = \big\{\ [\boldsymbol{B}_t\boldsymbol{X},\ -\boldsymbol{X}\boldsymbol{A}_t]\ :\ \boldsymbol{X} \in \mathbb{R}^{r\times r}\ \big\},$$

which is an r^2-dimensional linear subspace of $[\mathbb{R}^{m\times r}, \mathbb{R}^{r\times n}]$. Consequently, $\operatorname{rank}(J_{\mathcal{G}}) = (m + n)r - r^2$.

Proof. We prove the three statements in turn: (i) the indicated set is contained in $\ker(J_{\mathcal{G}})$; (ii) every kernel element has this form; (iii) the dimension equals r^2. The rank statement then follows by the rank–nullity theorem applied to $J_{\mathcal{G}} : [\mathbb{R}^{m\times r}, \mathbb{R}^{r\times n}] \to \mathbb{R}^{m\times n}$, whose domain has dimension $(m + n)r$.

(i) Inclusion. For any $\boldsymbol{X} \in \mathbb{R}^{r\times r}$, by the definition of $J_{\mathcal{G}}$ in (1),

$$J_{\mathcal{G}}\big[\boldsymbol{B}_t\boldsymbol{X},\ -\boldsymbol{X}\boldsymbol{A}_t\big] = (\boldsymbol{B}_t\boldsymbol{X})\boldsymbol{A}_t + \boldsymbol{B}_t(-\boldsymbol{X}\boldsymbol{A}_t) = \boldsymbol{B}_t\boldsymbol{X}\boldsymbol{A}_t - \boldsymbol{B}_t\boldsymbol{X}\boldsymbol{A}_t = \boldsymbol{0}.$$

Hence $[\boldsymbol{B}_t\boldsymbol{X},\ -\boldsymbol{X}\boldsymbol{A}_t] \in \ker(J_{\mathcal{G}})$ for every $\boldsymbol{X}$.

(ii) Reverse inclusion. Suppose $[\boldsymbol{P}, \boldsymbol{Q}] \in \ker(J_{\mathcal{G}})$, i.e., $\boldsymbol{P}\boldsymbol{A}_t + \boldsymbol{B}_t\boldsymbol{Q} = \boldsymbol{0}$, equivalently

$$\boldsymbol{P}\boldsymbol{A}_t = -\boldsymbol{B}_t\boldsymbol{Q}. \tag{15}$$

Since $\boldsymbol{B}_t$ has column rank r and $\boldsymbol{A}_t$ has row rank r, define their one-sided pseudoinverses

$$\boldsymbol{B}_t^+ := (\boldsymbol{B}_t^\top\boldsymbol{B}_t)^{-1}\boldsymbol{B}_t^\top \in \mathbb{R}^{r\times m}, \qquad \boldsymbol{A}_t^+ := \boldsymbol{A}_t^\top(\boldsymbol{A}_t\boldsymbol{A}_t^\top)^{-1} \in \mathbb{R}^{n\times r},$$

satisfying $\boldsymbol{B}_t^+\boldsymbol{B}_t = \boldsymbol{I}_r$ and $\boldsymbol{A}_t\boldsymbol{A}_t^+ = \boldsymbol{I}_r$. Set $\boldsymbol{X} := \boldsymbol{B}_t^+\boldsymbol{P} \in \mathbb{R}^{r\times r}$.

Left-multiplying (15) by $\boldsymbol{B}_t^+$ gives $\boldsymbol{X}\boldsymbol{A}_t = -\boldsymbol{Q}$, i.e.,

$$\boldsymbol{Q} = -\boldsymbol{X}\boldsymbol{A}_t.$$

Substituting back into (15) and right-multiplying by $\boldsymbol{A}_t^+$ gives

$$\boldsymbol{P} = \boldsymbol{P}\boldsymbol{A}_t\boldsymbol{A}_t^+ = -\boldsymbol{B}_t\boldsymbol{Q}\boldsymbol{A}_t^+ = \boldsymbol{B}_t\boldsymbol{X}\boldsymbol{A}_t\boldsymbol{A}_t^+ = \boldsymbol{B}_t\boldsymbol{X}.$$

Therefore $[\boldsymbol{P}, \boldsymbol{Q}] = [\boldsymbol{B}_t\boldsymbol{X},\ -\boldsymbol{X}\boldsymbol{A}_t]$, as claimed.

(iii) Dimension. The map $\Phi : \mathbb{R}^{r\times r} \to [\mathbb{R}^{m\times r}, \mathbb{R}^{r\times n}]$ defined by $\Phi(\boldsymbol{X}) = [\boldsymbol{B}_t\boldsymbol{X},\ -\boldsymbol{X}\boldsymbol{A}_t]$ is linear. It is injective: if $\Phi(\boldsymbol{X}) = \mathbf{0}$ then $\boldsymbol{B}_t\boldsymbol{X} = \mathbf{0}$, and the column-rank condition on $\boldsymbol{B}_t$ gives $\boldsymbol{X} = \mathbf{0}$. Hence $\dim(\operatorname{im}\Phi) = \dim(\mathbb{R}^{r\times r}) = r^2$. By (i)–(ii), $\operatorname{im}\Phi = \ker(J_{\mathcal{G}})$, so $\dim\ker(J_{\mathcal{G}}) = r^2$.

Rank. By the rank–nullity theorem,

$$\operatorname{rank}(J_{\mathcal{G}}) \;=\; \dim\left([\mathbb{R}^{m\times r}, \mathbb{R}^{r\times n}]\right) - \dim\ker(J_{\mathcal{G}}) \;=\; (m+n)r - r^2.$$

Equivalently, $\operatorname{rank}(J_{\mathcal{G}}) = \dim\mathcal{M}_r$, the dimension of the rank-r manifold at $\boldsymbol{W}_t = \boldsymbol{B}_t\boldsymbol{A}_t$, consistent with $\operatorname{im}(J_{\mathcal{G}}) = \mathbb{T}_t$ (Proposition C.6). □

C.3 Optimality of $\mathcal{H}_t$-Projection (Solution 1)

Proposition C.3 (Common $\boldsymbol{W}$-update across the affine solution set)**.** *Let $\mathcal{H}_t : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ be a symmetric positive-definite operator, inducing the inner product $\langle \boldsymbol{X}, \boldsymbol{Y}\rangle_{\mathcal{H}_t} := \langle \mathcal{H}_t\boldsymbol{X}, \boldsymbol{Y}\rangle$ on $\mathbb{R}^{m\times n}$. Let $\mathbb{T}_t := \operatorname{range}(J_{\mathcal{G}}) \subset \mathbb{R}^{m\times n}$ and $\widetilde{\boldsymbol{G}}_t := \mathcal{H}_t^{-1}\boldsymbol{G}_t$. Then every $[\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}]$ in the affine solution set of* (10) *satisfies*

$$J_{\mathcal{G}}[\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}] = \widetilde{\mathcal{P}}_{\mathbb{T}_t}(\widetilde{\boldsymbol{G}}_t),$$

where $\widetilde{\mathcal{P}}_{\mathbb{T}_t}$ is the $\mathcal{H}_t$-orthogonal projector onto $\mathbb{T}_t$, characterized by $\langle \widetilde{\mathcal{P}}_{\mathbb{T}_t}(\boldsymbol{X}) - \boldsymbol{X}, \boldsymbol{Y}\rangle_{\mathcal{H}_t} = 0$ for all $\boldsymbol{X} \in \mathbb{R}^{m\times n}, \boldsymbol{Y} \in \mathbb{T}_t$.

Proof. Equip $V := \mathbb{R}^{m\times n}$ with the inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}_t}$. Since $\mathcal{H}_t$ is SPD, this is a genuine inner product, and V is a finite-dimensional Hilbert space. The subspace $\mathbb{T}_t$ is finite-dimensional, hence closed in V, and admits a unique $\mathcal{H}_t$-orthogonal decomposition $V = \mathbb{T}_t \oplus \mathbb{T}_t^{\perp_{\mathcal{H}_t}}$, defining $\widetilde{\mathcal{P}}_{\mathbb{T}_t}$.

Let $[\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}]$ be any solution of (10). Using $J_{\mathcal{G}}^*(\boldsymbol{G}_t) = J_{\mathcal{G}}^*\mathcal{H}_t\widetilde{\boldsymbol{G}}_t$, the equation rewrites as $J_{\mathcal{G}}^*\mathcal{H}_t(J_{\mathcal{G}}[\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}] - \widetilde{\boldsymbol{G}}_t) = 0$, i.e. $J_{\mathcal{G}}[\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}] - \widetilde{\boldsymbol{G}}_t \in \mathbb{T}_t^{\perp_{\mathcal{H}_t}}$ (since $\ker(J_{\mathcal{G}}^*\mathcal{H}_t) = \mathbb{T}_t^{\perp_{\mathcal{H}_t}}$ by adjointness). Decomposing $\widetilde{\boldsymbol{G}}_t = \widetilde{\mathcal{P}}_{\mathbb{T}_t}(\widetilde{\boldsymbol{G}}_t) + \widetilde{\mathcal{P}}_{\mathbb{T}_t}^{\perp}(\widetilde{\boldsymbol{G}}_t)$, this gives $J_{\mathcal{G}}[\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}] - \widetilde{\mathcal{P}}_{\mathbb{T}_t}(\widetilde{\boldsymbol{G}}_t) \in \mathbb{T}_t^{\perp_{\mathcal{H}_t}}$. But the left-hand side also lies in $\mathbb{T}_t$ (both summands do), and $\mathbb{T}_t \cap \mathbb{T}_t^{\perp_{\mathcal{H}_t}} = \{0\}$, so $J_{\mathcal{G}}[\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}] = \widetilde{\mathcal{P}}_{\mathbb{T}_t}(\widetilde{\boldsymbol{G}}_t)$. □

C.4 Orthogonal Projection to Tangent Space

In this subsection, we derive the orthogonal projection onto the tangent space under both the standard metric and the weighted metric. The specific forms of $\boldsymbol{L}_t$ and $\boldsymbol{R}_t$ are presented here and will not be repeated in subsequent propositions and proofs. For the sake of simplicity, the subscript t will be omitted in this subsection.

$$\begin{aligned}
\boldsymbol{L}_t &= \operatorname{diag}(\boldsymbol{l}_t/\sqrt{\|\boldsymbol{l}_t\|_1}) \ \text{ with } \ \boldsymbol{l}_t = \beta_1\boldsymbol{l}_{t-1} + (1-\beta_1)\sum_{j=1}^{n}(\boldsymbol{G}_t \odot \boldsymbol{G}_t)_{i,j}, \\
\boldsymbol{R}_t &= \operatorname{diag}(\boldsymbol{r}_t/\sqrt{\|\boldsymbol{r}_t\|_1}) \ \text{ with } \ \boldsymbol{r}_t = \beta_2\boldsymbol{r}_{t-1} + (1-\beta_2)\sum_{i=1}^{m}(\boldsymbol{G}_t \odot \boldsymbol{G}_t)_{i,j},
\end{aligned} \tag{16}$$

where $\odot$ denotes the Hadamard (elementwise) product and $\boldsymbol{G}_t = \nabla\mathcal{L}(\boldsymbol{W}_0 + \boldsymbol{W}_t)$.

Proposition C.4 (Orthogonal Projection to Tangent Space Under the Standard Metric)**.** *Let $\boldsymbol{W} \in \mathcal{M}_r$ be a rank-r matrix with a low-rank decomposition $\boldsymbol{W} = \boldsymbol{B}\boldsymbol{A}$, where $\boldsymbol{B} \in \mathbb{R}^{m\times r}, \boldsymbol{A} \in \mathbb{R}^{r\times n}$. Denote by $\mathbb{T}_{\boldsymbol{W}}$ the tangent space of the smooth manifold $\mathcal{M}_r$ at the point $\boldsymbol{W}$. Then, the orthogonal projection of any matrix $\boldsymbol{Z} \in \mathbb{R}^{m\times n}$ onto $\mathbb{T}_{\boldsymbol{W}}$ is given by*

$$\mathcal{P}_{\mathbb{T}_{\boldsymbol{W}}}(\boldsymbol{Z}) = \boldsymbol{B}(\boldsymbol{B}^\top\boldsymbol{B})^{-1}\boldsymbol{B}^\top\boldsymbol{Z} + \boldsymbol{Z}\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{A}^\top)^{-1}\boldsymbol{A} - \boldsymbol{B}(\boldsymbol{B}^\top\boldsymbol{B})^{-1}\boldsymbol{B}^\top\boldsymbol{Z}\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{A}^\top)^{-1}\boldsymbol{A}.$$

Proof. Suppose $\boldsymbol{W}$ has a compact singular value decomposition, given by $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, where $\boldsymbol{U} \in \mathbb{R}^{m\times r}, \boldsymbol{\Sigma} \in \mathbb{R}^{r\times r}, \boldsymbol{V} \in \mathbb{R}^{n\times r}$. Then the tangent space $\mathbb{T}_{\boldsymbol{W}}$ at $\boldsymbol{W}$ is characterized as

$$\mathbb{T}_{\boldsymbol{W}} = \{\boldsymbol{U}\boldsymbol{M}^\top + \boldsymbol{N}\boldsymbol{V}^\top, \text{for } \boldsymbol{M} \in \mathbb{R}^{n\times r}, \boldsymbol{N} \in \mathbb{R}^{m\times r}\}.$$

Therefore, the orthogonal projection of $\boldsymbol{Z}$ onto $\mathbb{T}_{\boldsymbol{W}}$ is known to be [35]

$$\mathcal{P}_{\mathbb{T}_{\boldsymbol{W}}}(\boldsymbol{Z}) = \boldsymbol{U}\boldsymbol{U}^\top\boldsymbol{Z} + \boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^\top - \boldsymbol{U}\boldsymbol{U}^\top\boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^\top. \tag{17}$$

Since the columns of $\boldsymbol{B}$ and $\boldsymbol{U}$ span the same column space (i.e., the column space of $\boldsymbol{W}$), then there exists an invertible matrix $\boldsymbol{S} \in \mathbb{R}^{r\times r}$ such that $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{S}$ and $\boldsymbol{U} = \boldsymbol{B}\boldsymbol{S}^{-1}$. Using this relation, we have

$$\boldsymbol{U}^\top\boldsymbol{U} = (\boldsymbol{B}\boldsymbol{S}^{-1})^\top\boldsymbol{B}\boldsymbol{S}^{-1} = \boldsymbol{S}^{-\top}(\boldsymbol{B}^\top\boldsymbol{B})\boldsymbol{S}^{-1}.$$

Since $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}_r$, it follows that

$$\boldsymbol{S}^{-\top}(\boldsymbol{B}^\top\boldsymbol{B})\boldsymbol{S}^{-1} = \boldsymbol{I}_r \implies \boldsymbol{B}^\top\boldsymbol{B} = \boldsymbol{S}^\top\boldsymbol{S}.$$

Using this, we compute $\boldsymbol{U}\boldsymbol{U}^\top$

$$\boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{S}^{-\top}\boldsymbol{B}^\top = \boldsymbol{B}(\boldsymbol{S}^\top\boldsymbol{S})^{-1}\boldsymbol{B}^\top = \boldsymbol{B}(\boldsymbol{B}^\top\boldsymbol{B})^{-1}\boldsymbol{B}^\top \tag{18}$$

Similarly, since the rows of $\boldsymbol{A}$ and the columns of $\boldsymbol{V}$ span the same row space (i.e., the row space of $\boldsymbol{W}$), there exists an invertible matrix $\boldsymbol{Q} \in \mathbb{R}^{r\times r}$ such that $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{V}^\top$ and $\boldsymbol{V}^\top = \boldsymbol{Q}^{-1}\boldsymbol{A}$. Further, using $\boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I}_r$, we obtain

$$\boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{Q}^{-1}(\boldsymbol{A}\boldsymbol{A}^\top)\boldsymbol{Q}^{-\top} = \boldsymbol{I}_r,$$

hence $\boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{Q}\boldsymbol{Q}^\top$ and

$$\boldsymbol{V}\boldsymbol{V}^\top = \boldsymbol{A}^\top\boldsymbol{Q}^{-\top}\boldsymbol{Q}^{-1}\boldsymbol{A} = \boldsymbol{A}^\top(\boldsymbol{Q}\boldsymbol{Q}^\top)^{-1}\boldsymbol{A} = \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{A}^\top)^{-1}\boldsymbol{A} \tag{19}$$

Substituting (18) and (19) into (17) yields

$$\mathcal{P}_{\mathbb{T}_{\boldsymbol{W}}}(\boldsymbol{Z}) = \boldsymbol{B}(\boldsymbol{B}^\top\boldsymbol{B})^{-1}\boldsymbol{B}^\top\boldsymbol{Z} + \boldsymbol{Z}\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{A}^\top)^{-1}\boldsymbol{A} - \boldsymbol{B}(\boldsymbol{B}^\top\boldsymbol{B})^{-1}\boldsymbol{B}^\top\boldsymbol{Z}\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{A}^\top)^{-1}\boldsymbol{A}.$$

□

Proposition C.5 (Orthogonal Projection onto the Tangent Space Under the Weighted Metric)**.** *Let $\boldsymbol{W} \in \mathcal{M}_r$ have a low-rank decomposition $\boldsymbol{W} = \boldsymbol{B}\boldsymbol{A}$, where $\boldsymbol{B} \in \mathbb{R}^{m\times r}, \boldsymbol{A} \in \mathbb{R}^{r\times n}$. Denote the tangent space of the Riemannian manifold $\mathcal{M}_r$ at the point $\boldsymbol{W}$ as $\mathbb{T}_W$. The weighted metric is defined as $\langle \boldsymbol{Y}, \boldsymbol{Z}\rangle_{\boldsymbol{H}} = \langle \boldsymbol{L}^{\frac{1}{2}}\boldsymbol{Y}\boldsymbol{R}^{\frac{1}{2}}, \boldsymbol{Z}\rangle$ for any $\boldsymbol{Y}, \boldsymbol{Z} \in \mathbb{R}^{m\times n}$. Then, the orthogonal projection of any matrix $\boldsymbol{Z} \in \mathbb{R}^{m\times n}$ onto $\mathbb{T}_W$ under the weighted metric is given by*

$$\begin{aligned}\mathcal{P}_{\mathbb{T}_{\boldsymbol{W}}}(\boldsymbol{Z}) = {} & \boldsymbol{B}(\boldsymbol{B}^\top\boldsymbol{L}^{\frac{1}{2}}\boldsymbol{B})^{-1}\boldsymbol{B}^\top\boldsymbol{L}^{\frac{1}{2}}\boldsymbol{Z} + \boldsymbol{Z}\boldsymbol{R}^{\frac{1}{2}}\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{R}^{\frac{1}{2}}\boldsymbol{A}^\top)^{-1}\boldsymbol{A} \\ & - \boldsymbol{B}(\boldsymbol{B}^\top\boldsymbol{L}^{\frac{1}{2}}\boldsymbol{B})^{-1}\boldsymbol{B}^\top\boldsymbol{L}^{\frac{1}{2}}\boldsymbol{Z}\boldsymbol{R}^{\frac{1}{2}}\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{R}^{\frac{1}{2}}\boldsymbol{A}^\top)^{-1}\boldsymbol{A}.\end{aligned}$$

Proof. This proof is inspired by [3]. Here, we briefly provide a sketch of the proof.

(i) *The new orthonormal basis under the weighted metric.* Let $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ be a compact SVD with $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_r] \in \mathbb{R}^{m\times r}, \boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_r] \in \mathbb{R}^{n\times r}$. Normalize the singular vectors under the weighted vector

$$\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\boldsymbol{L}^{\frac{1}{2}}} = \langle \boldsymbol{L}^{\frac{1}{2}}\boldsymbol{x}, \boldsymbol{y}\rangle \text{ in } \mathbb{R}^m \quad \text{and} \quad \langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\boldsymbol{R}^{\frac{1}{2}}} = \langle \boldsymbol{R}^{\frac{1}{2}}\boldsymbol{x}, \boldsymbol{y}\rangle \text{ in } \mathbb{R}^n$$

to obtain

$$\begin{aligned}\widetilde{\boldsymbol{U}} &= \boldsymbol{U}(\boldsymbol{U}^\top\boldsymbol{L}^{\frac{1}{2}}\boldsymbol{U})^{-\frac{1}{2}} \coloneqq [\tilde{\boldsymbol{u}}_1, \tilde{\boldsymbol{u}}_2, \cdots, \tilde{\boldsymbol{u}}_r] \in \mathbb{R}^{m\times r}, \\ \widetilde{\boldsymbol{V}} &= \boldsymbol{V}(\boldsymbol{V}^\top\boldsymbol{R}^{\frac{1}{2}}\boldsymbol{V})^{-\frac{1}{2}} \coloneqq [\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2, \cdots, \tilde{\boldsymbol{v}}_r] \in \mathbb{R}^{n\times r}.\end{aligned}$$

Next, we extend $\widetilde{\boldsymbol{U}}$ and $\widetilde{\boldsymbol{V}}$ to full orthonormal bases of $(\mathbb{R}^m, \langle\cdot,\cdot\rangle_{\boldsymbol{L}^{\frac{1}{2}}})$ and $(\mathbb{R}^n, \langle\cdot,\cdot\rangle_{\boldsymbol{R}^{\frac{1}{2}}})$ respectively. Then, an orthonormal basis of $\mathbb{T}_{\boldsymbol{W}}$ with respect to $\langle\cdot,\cdot\rangle_{\boldsymbol{H}_t}$ is $\{\tilde{\boldsymbol{u}}_i\tilde{\boldsymbol{v}}_j^\top\}_{\min\{i,j\}\le r}$.

(ii) *Orthogonal projection represented by the new orthonormal basis.* Using the orthonormal bases $\widetilde{\boldsymbol{U}}$ and $\widetilde{\boldsymbol{V}}$, the projection of $\boldsymbol{Z}$ onto $\mathbb{T}_{\boldsymbol{W}}$ is expressed as:

$$\begin{aligned}\widetilde{\mathcal{P}}_{\mathbb{T}_{\boldsymbol{W}}}(\boldsymbol{Z}) &= \sum_{(i,j):\min\{i,j\}\le r} \langle \boldsymbol{Z}, \tilde{\boldsymbol{u}}_i\tilde{\boldsymbol{v}}_j^\top\rangle_{\boldsymbol{H}_t}\cdot\tilde{\boldsymbol{u}}_i\tilde{\boldsymbol{v}}_j^\top = \sum_{(i,j):\min\{i,j\}\le r} \langle \boldsymbol{L}^{\frac{1}{2}}\boldsymbol{Z}\boldsymbol{R}^{\frac{1}{2}}, \tilde{\boldsymbol{u}}_i\tilde{\boldsymbol{v}}_j^\top\rangle\cdot\tilde{\boldsymbol{u}}_i\tilde{\boldsymbol{v}}_j^\top \\ &= \sum_{(i,j):\min\{i,j\}\le r} \tilde{\boldsymbol{u}}_i^\top\boldsymbol{L}^{\frac{1}{2}}\boldsymbol{Z}\boldsymbol{R}^{\frac{1}{2}}\tilde{\boldsymbol{v}}_j\cdot\tilde{\boldsymbol{u}}_i\tilde{\boldsymbol{v}}_j^\top \\ &= \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{U}}^\top\boldsymbol{L}^{\frac{1}{2}}\boldsymbol{Z} + \boldsymbol{Z}\boldsymbol{R}^{\frac{1}{2}}\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^\top - \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{U}}^\top\boldsymbol{L}^{\frac{1}{2}}\boldsymbol{Z}\boldsymbol{R}^{\frac{1}{2}}\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^\top.\end{aligned}$$

(iii) *Express the basis projectors via factors $\boldsymbol{B}$ and $\boldsymbol{A}$.* Since $\boldsymbol{B}$ and $\boldsymbol{A}$ span the same spaces as $\boldsymbol{U}$ and $\boldsymbol{V}$, we derive

$$\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{U}}^\top = \boldsymbol{B}\Big(\boldsymbol{B}^\top \boldsymbol{L}^{\frac{1}{2}}\boldsymbol{B}\Big)^{-1}\boldsymbol{B}^\top, \quad \widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^\top = \boldsymbol{A}^\top\Big(\boldsymbol{A}\boldsymbol{R}^{\frac{1}{2}}\boldsymbol{A}^\top\Big)^{-1}\boldsymbol{A}.$$

Substituting these expressions into the formula for $\widetilde{\mathcal{P}}_{\mathbb{T}_{\boldsymbol{W}}}$, we obtain

$$\begin{aligned}\widetilde{\mathcal{P}}_{\mathbb{T}_{\boldsymbol{W}}}(\boldsymbol{Z}) = \boldsymbol{B}(\boldsymbol{B}^\top \boldsymbol{L}^{\frac{1}{2}}\boldsymbol{B})^{-1}\boldsymbol{B}^\top \boldsymbol{L}^{\frac{1}{2}}\boldsymbol{Z} + \boldsymbol{Z}\boldsymbol{R}^{\frac{1}{2}}\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{R}^{\frac{1}{2}}\boldsymbol{A}^\top)^{-1}\boldsymbol{A} \\ - \boldsymbol{B}(\boldsymbol{B}^\top \boldsymbol{L}^{\frac{1}{2}}\boldsymbol{B})^{-1}\boldsymbol{B}^\top \boldsymbol{L}^{\frac{1}{2}}\boldsymbol{Z}\boldsymbol{R}^{\frac{1}{2}}\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{R}^{\frac{1}{2}}\boldsymbol{A}^\top)^{-1}\boldsymbol{A}.\end{aligned}$$

□

Proposition C.6. *Suppose $\boldsymbol{W} \in \mathcal{M}_r$ has a low-rank decomposition $\boldsymbol{W} = \boldsymbol{B}\boldsymbol{A}$, where $\boldsymbol{B} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{A} \in \mathbb{R}^{r\times n}$. For any matrix $\boldsymbol{M} \in \mathbb{R}^{m\times r}$, $\boldsymbol{N} \in \mathbb{R}^{r\times n}$, the matrix $\boldsymbol{M}\boldsymbol{A} + \boldsymbol{B}\boldsymbol{N}$ lies in the tangent space $\mathbb{T}_{\boldsymbol{W}}$ of $\mathcal{M}_r$ at the point $\boldsymbol{W}$.*

Proof. Let $\boldsymbol{W} \in \mathcal{M}_r$ have a compact singular value decomposition $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, where $\boldsymbol{U} \in \mathbb{R}^{m\times r}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{r\times r}$, and $\boldsymbol{V} \in \mathbb{R}^{n\times r}$. By definition, the tangent space $\mathbb{T}_{\boldsymbol{W}}$ at $\boldsymbol{W}$ is given by

$$\mathbb{T}_{\boldsymbol{W}} = \{\boldsymbol{U}\boldsymbol{K}_1^\top + \boldsymbol{K}_2\boldsymbol{V}^\top | \boldsymbol{K}_1 \in \mathbb{R}^{n\times r}, \boldsymbol{K}_2 \in \mathbb{R}^{m\times r}\}.$$

Since $\boldsymbol{B}$ and $\boldsymbol{A}$ are low-rank factors of $\boldsymbol{W}$, there exist invertible matrices $\boldsymbol{S} \in \mathbb{R}^{r\times r}$ and $\boldsymbol{Q} \in \mathbb{R}^{r\times r}$ such that

$$\boldsymbol{B} = \boldsymbol{U}\boldsymbol{S}, \quad \boldsymbol{A} = \boldsymbol{Q}\boldsymbol{V}^\top.$$

Substituting these expressions, the matrix $\boldsymbol{M}\boldsymbol{A} + \boldsymbol{B}\boldsymbol{N}$ can be rewritten as

$$\boldsymbol{M}\boldsymbol{A} + \boldsymbol{B}\boldsymbol{N} = \boldsymbol{M}\boldsymbol{Q}\boldsymbol{V}^\top + \boldsymbol{U}\boldsymbol{S}\boldsymbol{N}.$$

The first term, $\boldsymbol{M}\boldsymbol{Q}\boldsymbol{V}^\top$, lies in span$(\boldsymbol{V}^\top)$, and the second term, $\boldsymbol{U}\boldsymbol{S}\boldsymbol{N}$, lies in span$(\boldsymbol{U})$. Thus, the sum $\boldsymbol{M}\boldsymbol{Q}\boldsymbol{V}^\top + \boldsymbol{U}\boldsymbol{S}\boldsymbol{N}$ lies in the tangent space $\mathbb{T}_{\boldsymbol{W}}$ by the definition of the tangent space. Then, it follows that $\boldsymbol{M}\boldsymbol{A} + \boldsymbol{B}\boldsymbol{N}$ is on the tangent space $\mathbb{T}_{\boldsymbol{W}}$. This completes the proof. □

C.5 Proof of Theorem 3.2 (closed-form factor update)

The proof has two stages: (i) solve (11) for $(\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t})$ in terms of any matrix $\boldsymbol{X}_t \in \mathbb{R}^{r\times r}$ (Lemma C.1); (ii) determine $\boldsymbol{X}_t$ by minimizing the $\mathcal{H}_t$-imbalance from Solution 2 (Lemma C.2). Substituting (ii) into (i) yields the closed form in Theorem 3.2.

Lemma C.1 ($\boldsymbol{X}_t$-parameterized factor solution)**.** *For problem* (11)*, the optimal $(\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t})$ are*

$$\boldsymbol{\Delta}_{\boldsymbol{B}_t}^{\text{opt}} = (\boldsymbol{I} - \widetilde{\boldsymbol{P}}_{B_t})\boldsymbol{L}_t^{-1/2}\boldsymbol{G}_{\boldsymbol{B}_t}(\boldsymbol{A}_t\boldsymbol{R}_t^{1/2}\boldsymbol{A}_t^\top)^{-1} - \boldsymbol{B}_t\boldsymbol{X}_t, \quad \boldsymbol{\Delta}_{\boldsymbol{A}_t}^{\text{opt}} = (\boldsymbol{B}_t^\top\boldsymbol{L}_t^{1/2}\boldsymbol{B}_t)^{-1}\boldsymbol{G}_{\boldsymbol{A}_t}\boldsymbol{R}_t^{-1/2} + \boldsymbol{X}_t\boldsymbol{A}_t,$$

where $\boldsymbol{X}_t \in \mathbb{R}^{r\times r}$ is arbitrary.

Proof of Lemma C.1. Define

$$\Gamma(\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}) := \frac{1}{2}\|\boldsymbol{\Delta}_{\boldsymbol{B}_t}\boldsymbol{A}_t + \boldsymbol{B}_t\boldsymbol{\Delta}_{\boldsymbol{A}_t} - \widetilde{\mathcal{P}}_{\mathbb{T}_t}(\boldsymbol{L}_t^{-\frac{1}{2}}\boldsymbol{G}_t\boldsymbol{R}_t^{-\frac{1}{2}})\|_{\mathcal{H}_t}^2.$$

Differentiating $\Gamma(\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t})$ with respect to $\boldsymbol{\Delta}_{\boldsymbol{B}_t}$ and $\boldsymbol{\Delta}_{\boldsymbol{A}_t}$ yields

$$\nabla_{\boldsymbol{\Delta}_{\boldsymbol{B}_t}}\Gamma(\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}) = \boldsymbol{L}_t^{\frac{1}{2}}\boldsymbol{\Delta}_{\boldsymbol{B}_t}(\boldsymbol{A}_t\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top) + \boldsymbol{L}_t^{\frac{1}{2}}\boldsymbol{B}_t\boldsymbol{\Delta}_{\boldsymbol{A}_t}\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top - \boldsymbol{G}_t\boldsymbol{A}_t^\top, \tag{20}$$

and

$$\nabla_{\boldsymbol{\Delta}_{\boldsymbol{A}_t}}\Gamma(\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}) = \boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}\boldsymbol{\Delta}_{\boldsymbol{B}_t}\boldsymbol{A}_t\boldsymbol{R}_t^{\frac{1}{2}} + \boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}\boldsymbol{B}_t\boldsymbol{\Delta}_{\boldsymbol{A}_t}\boldsymbol{R}_t^{\frac{1}{2}} - \boldsymbol{B}_t^\top\boldsymbol{G}_t. \tag{21}$$

Setting $\nabla_{\boldsymbol{\Delta}_{\boldsymbol{B}_t}}\Gamma(\boldsymbol{\Delta}_{\boldsymbol{B}_t}, \boldsymbol{\Delta}_{\boldsymbol{A}_t}) = \boldsymbol{0}$ and using the invertibility of $(\boldsymbol{A}_t\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top)$ and $\boldsymbol{L}_t^{\frac{1}{2}}$ gives

$$\boldsymbol{\Delta}_{\boldsymbol{B}_t} = \boldsymbol{L}_t^{-\frac{1}{2}}\boldsymbol{G}_t\boldsymbol{A}_t^\top(\boldsymbol{A}_t\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top)^{-1} - \boldsymbol{B}_t\boldsymbol{\Delta}_{\boldsymbol{A}_t}\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top(\boldsymbol{A}_t\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top)^{-1}. \tag{22}$$

Substituting (22) into $\nabla_{\boldsymbol{\Delta}_{\boldsymbol{A}_t}}\Gamma(\boldsymbol{\Delta}_{\boldsymbol{B}_t},\boldsymbol{\Delta}_{\boldsymbol{A}_t})=\mathbf{0}$ and using the invertibility of $\boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}\boldsymbol{B}_t$ and $\boldsymbol{R}_t^{\frac{1}{2}}$ yields

$$\boldsymbol{\Delta}_{\boldsymbol{A}_t}[\boldsymbol{I}-\widetilde{\boldsymbol{Q}}_{A_t}]=(\boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}\boldsymbol{B}_t)^{-1}\boldsymbol{B}_t^\top\boldsymbol{G}_t\boldsymbol{R}_t^{-\frac{1}{2}}[\boldsymbol{I}-\widetilde{\boldsymbol{Q}}_{A_t}],$$

where $\widetilde{\boldsymbol{Q}}_{A_t}=\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top(\boldsymbol{A}_t\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top)^{-1}\boldsymbol{A}_t$, which is the projection matrix onto the row space of $\boldsymbol{A}_t$. Since $\boldsymbol{I}-\widetilde{\boldsymbol{Q}}_{A_t}$ is the residual maker matrix, then a general solution is

$$\boldsymbol{\Delta}_{\boldsymbol{A}_t}^{\text{opt}}=(\boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}\boldsymbol{B}_t)^{-1}\boldsymbol{B}_t^\top\boldsymbol{G}_t\boldsymbol{R}_t^{-\frac{1}{2}}+\boldsymbol{X}_t\boldsymbol{A}_t,$$

with arbitrary matrix $\boldsymbol{X}_t\in\mathbb{R}^{r\times r}$. Plugging this $\boldsymbol{\Delta}_{\boldsymbol{A}_t}$ back into (22) gives

$$\boldsymbol{\Delta}_{\boldsymbol{B}_t}^{\text{opt}}=[\boldsymbol{I}-\widetilde{\boldsymbol{P}}_{B_t}]\boldsymbol{L}_t^{-\frac{1}{2}}\boldsymbol{G}_t\boldsymbol{A}_t^\top(\boldsymbol{A}_t\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top)^{-1}-\boldsymbol{B}_t\boldsymbol{X}_t,$$

where $\widetilde{\boldsymbol{P}}_{B_t}=\boldsymbol{B}_t(\boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}\boldsymbol{B}_t)^{-1}\boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}$, which is the projection matrix onto the column space of $\boldsymbol{B}_t$. □

Lemma C.2 ($\mathcal{H}_t$-balance closed form)**.** *The minimizer of $\frac{1}{2}\|\boldsymbol{\Delta}_{\boldsymbol{B}_t}\boldsymbol{A}_t-\boldsymbol{B}_t\boldsymbol{\Delta}_{\boldsymbol{A}_t}\|_{\mathcal{H}_t}^2$ over $\boldsymbol{X}_t\in\mathbb{R}^{r\times r}$ (with $\boldsymbol{\Delta}_{\boldsymbol{B}_t},\boldsymbol{\Delta}_{\boldsymbol{A}_t}$ given by Lemma C.1) is*

$$\boldsymbol{X}_t^{\text{opt}}=-\frac{1}{2}(\boldsymbol{B}_t^\top\boldsymbol{L}_t^{1/2}\boldsymbol{B}_t)^{-1}\boldsymbol{B}_t^\top\boldsymbol{G}_t\boldsymbol{A}_t^\top(\boldsymbol{A}_t\boldsymbol{R}_t^{1/2}\boldsymbol{A}_t^\top)^{-1}.$$

Substituting $\boldsymbol{X}_t^{\text{opt}}$ into Lemma C.1 yields the closed form in Theorem 3.2.

Proof of Lemma C.2. Let the objective function be $\Psi(\boldsymbol{X}_t)=\frac{1}{2}\|\boldsymbol{\Delta}_{\boldsymbol{B}_t}\boldsymbol{A}_t-\boldsymbol{B}_t\boldsymbol{\Delta}_{\boldsymbol{A}_t}\|_{\mathcal{H}_t}^2$. To minimize $\Psi(\boldsymbol{X}_t)$, we compute its gradient with respect to $\boldsymbol{X}_t$,

$$\nabla_{\boldsymbol{X}_t}\Psi(\boldsymbol{X}_t)=-2\boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}(\boldsymbol{\Delta}_{\boldsymbol{B}_t}\boldsymbol{A}_t-\boldsymbol{B}_t\boldsymbol{\Delta}_{\boldsymbol{A}_t})\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}^\top.$$

Substituting the expressions for $\boldsymbol{\Delta}_{\boldsymbol{B}_t}$ and $\boldsymbol{\Delta}_{\boldsymbol{A}_t}$ from Lemma C.1, we have

$$\begin{aligned}\nabla_{\boldsymbol{X}_t}\Psi(\boldsymbol{X}_t)=-2\boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}\Bigg(&[\boldsymbol{I}-\boldsymbol{B}_t(\boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}\boldsymbol{B}_t)^{-1}\boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}]\boldsymbol{L}_t^{-\frac{1}{2}}\boldsymbol{G}_t\boldsymbol{A}_t^\top(\boldsymbol{A}_t\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top)^{-1}\boldsymbol{A}_t\\&-\boldsymbol{B}_t(\boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}\boldsymbol{B}_t)^{-1}\boldsymbol{B}_t^\top\boldsymbol{G}_t\boldsymbol{R}^{-\frac{1}{2}}-2\boldsymbol{B}_t\boldsymbol{X}_t\boldsymbol{A}_t\Bigg)\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}^\top\\=2\boldsymbol{B}_t^\top\boldsymbol{G}_t\boldsymbol{A}_t^\top&+4(\boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}\boldsymbol{B}_t)\boldsymbol{X}_t(\boldsymbol{A}_t\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top).\end{aligned}$$

Setting $\nabla_{\boldsymbol{X}_t}\Psi(\boldsymbol{X}_t)=\mathbf{0}$, we obtain

$$-\boldsymbol{B}_t^\top\boldsymbol{G}_t\boldsymbol{A}_t^\top=2(\boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}\boldsymbol{B}_t)\boldsymbol{X}_t(\boldsymbol{A}_t\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top).$$

Since $\boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}\boldsymbol{B}_t$ and $\boldsymbol{A}_t\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top$ are invertible, we solve for $\boldsymbol{X}_t$ as

$$\boldsymbol{X}_t^{\text{opt}}=-\frac{1}{2}(\boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}\boldsymbol{B}_t)^{-1}\boldsymbol{B}_t^\top\boldsymbol{G}_t\boldsymbol{A}_t^\top(\boldsymbol{A}_t\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top)^{-1}.$$

This proves Lemma C.2. Substituting $\boldsymbol{X}_t^{\text{opt}}$ into Lemma C.1 and simplifying $-\boldsymbol{B}_t\boldsymbol{X}_t=\frac{1}{2}\widetilde{\boldsymbol{P}}_{B_t}\boldsymbol{L}_t^{-1/2}\boldsymbol{G}_t\boldsymbol{A}_t^\top(\boldsymbol{A}_t\boldsymbol{R}_t^{1/2}\boldsymbol{A}_t^\top)^{-1}$ (and the symmetric expression for $\boldsymbol{X}_t\boldsymbol{A}_t$) gives the closed form in Theorem 3.2, where $\boldsymbol{G}_{\boldsymbol{B}_t}=\boldsymbol{G}_t\boldsymbol{A}_t^\top$ and $\boldsymbol{G}_{\boldsymbol{A}_t}=\boldsymbol{B}_t^\top\boldsymbol{G}_t$ are the chain-rule low-rank gradients. □

D Algorithm

In this section, we provide the complete algorithm of FILO-Adafactor.

E Computational and Memory Complexity Analysis of FILO-Adafactor

The update rule of FILO-Adafactor is given by

$$\boldsymbol{\Delta}_{\boldsymbol{A}_t}=(\boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}\boldsymbol{B}_t)^{-1}\underbrace{\boldsymbol{B}_t^\top\boldsymbol{G}_t}_{\boldsymbol{G}_{\boldsymbol{A}_t}}\boldsymbol{R}_t^{-\frac{1}{2}}\left[\boldsymbol{I}-\frac{1}{2}\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top(\boldsymbol{A}_t\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top)^{-1}\boldsymbol{A}_t\right],$$

$$\boldsymbol{\Delta}_{\boldsymbol{B}_t}=\left[\boldsymbol{I}-\frac{1}{2}\boldsymbol{B}_t(\boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}\boldsymbol{B}_t)^{-1}\boldsymbol{B}_t^\top\boldsymbol{L}_t^{\frac{1}{2}}\right]\boldsymbol{L}_t^{-\frac{1}{2}}\underbrace{\boldsymbol{G}_t\boldsymbol{A}_t^\top}_{\boldsymbol{G}_{\boldsymbol{B}_t}}(\boldsymbol{A}_t\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top)^{-1}.$$

Algorithm 1 FILO-Adafactor for fine-tuning.

Initialize $\boldsymbol{B}_1 = \boldsymbol{0}_{m\times r}$, $\boldsymbol{A}_1 = \text{Kaiming uniform}_{r\times n}$, $\boldsymbol{l}_0 = \boldsymbol{0}_m, \boldsymbol{r}_0 = \boldsymbol{0}_n$, $\epsilon = 1e-6$.
for $t = 1, \cdots, T$ **do**
  $\boldsymbol{l}_t = \beta_1 \boldsymbol{l}_{t-1} + (1-\beta_1)\sum_{j=1}^{n}(\boldsymbol{G}_t \odot \boldsymbol{G}_t)_{i,j}$, $\boldsymbol{L}_t = \text{diag}(\boldsymbol{l}_t / \sqrt{\|\boldsymbol{l}_t\|_1})$.
  $\boldsymbol{r}_t = \beta_2 \boldsymbol{r}_{t-1} + (1-\beta_2)\sum_{i=1}^{m}(\boldsymbol{G}_t \odot \boldsymbol{G}_t)_{i,j}$, $\boldsymbol{R}_t = \text{diag}(\boldsymbol{r}_t / \sqrt{\|\boldsymbol{r}_t\|_1})$.
  $\boldsymbol{\Delta}_{\boldsymbol{B}_t} = \left[\boldsymbol{I} - \frac{1}{2}\boldsymbol{B}_t(\boldsymbol{B}_t^\top \boldsymbol{L}_t^{\frac{1}{2}} \boldsymbol{B}_t)^{-1}\boldsymbol{B}_t^\top \boldsymbol{L}_t^{\frac{1}{2}}\right]\boldsymbol{L}_t^{-\frac{1}{2}}\boldsymbol{G}_{\boldsymbol{B}_t}(\boldsymbol{A}_t \boldsymbol{R}_t^{\frac{1}{2}} \boldsymbol{A}_t^\top)^{-1}$.
  $\boldsymbol{\Delta}_{\boldsymbol{A}_t} = (\boldsymbol{B}_t^\top \boldsymbol{L}_t^{\frac{1}{2}} \boldsymbol{B}_t)^{-1}\boldsymbol{G}_{\boldsymbol{A}_t}\boldsymbol{R}_t^{-\frac{1}{2}}\left[\boldsymbol{I} - \frac{1}{2}\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top(\boldsymbol{A}_t \boldsymbol{R}_t^{\frac{1}{2}} \boldsymbol{A}_t^\top)^{-1}\boldsymbol{A}_t\right]$.
  $\boldsymbol{B}_{t+1} = \boldsymbol{B}_t - \eta_t \boldsymbol{\Delta}_{\boldsymbol{B}_t}$, $\boldsymbol{A}_{t+1} = \boldsymbol{A}_t - \eta_t \boldsymbol{\Delta}_{\boldsymbol{A}_t}$.
end for
Note: factor gradients $\boldsymbol{G}_{\boldsymbol{B}_t} = \partial \mathcal{L}/\partial \boldsymbol{B}_t$ and $\boldsymbol{G}_{\boldsymbol{A}_t} = \partial \mathcal{L}/\partial \boldsymbol{A}_t$ are obtained from autograd. The weight-space gradient $\boldsymbol{G}_t$ can be obtained via a backward hook on $\boldsymbol{W}$. Add $\epsilon \boldsymbol{I}$ to matrix $\boldsymbol{B}_t^\top \boldsymbol{L}_t^{\frac{1}{2}} \boldsymbol{B}_t$ if it is not invertible.

Algorithm 2 FILO-Adafactor with Momentum (FILO-AdafactorM) for fine-tuning.

Initialize moment $\boldsymbol{M}_{\boldsymbol{B}_0} = \boldsymbol{0}_{m\times r}$, $\boldsymbol{M}_{\boldsymbol{A}_0} = \boldsymbol{0}_{r\times n}$, $\boldsymbol{B}_1 = \boldsymbol{0}_{m\times r}$, $\boldsymbol{A}_1 = \text{Kaiming uniform}_{r\times n}$; $\boldsymbol{l}_0 = \boldsymbol{0}_m, \boldsymbol{r}_0 = \boldsymbol{0}_n$, weight decay λ, coefficients $\beta_1 = \beta_2$, and β_3, $\epsilon = 1e-6$.
for $t = 1, \cdots, T$ **do**
  $\boldsymbol{l}_t = \beta_1 \boldsymbol{l}_{t-1} + (1-\beta_1)\sum_{j=1}^{n}(\boldsymbol{G}_t \odot \boldsymbol{G}_t)_{i,j}$, $\boldsymbol{L}_t = \text{diag}(\boldsymbol{l}_t / \sqrt{\|\boldsymbol{l}_t\|_1})$.
  $\boldsymbol{r}_t = \beta_2 \boldsymbol{r}_{t-1} + (1-\beta_2)\sum_{i=1}^{m}(\boldsymbol{G}_t \odot \boldsymbol{G}_t)_{i,j}$, $\boldsymbol{R}_t = \text{diag}(\boldsymbol{r}_t / \sqrt{\|\boldsymbol{r}_t\|_1})$.
  $\boldsymbol{M}_{\boldsymbol{B}_t} = \beta_3 \boldsymbol{M}_{\boldsymbol{B}_{t-1}} + (1-\beta_3)\boldsymbol{G}_{\boldsymbol{B}_t}$.
  $\boldsymbol{M}_{\boldsymbol{A}_t} = \beta_3 \boldsymbol{M}_{\boldsymbol{A}_{t-1}} + (1-\beta_3)\boldsymbol{G}_{\boldsymbol{A}_t}$.
  $\boldsymbol{\Delta}_{\boldsymbol{B}_t} = \left[\boldsymbol{I} - \frac{1}{2}\boldsymbol{B}_t\left(\boldsymbol{B}_t^\top \boldsymbol{L}_t^{\frac{1}{2}} \boldsymbol{B}_t\right)^{-1}\boldsymbol{B}_t^\top \boldsymbol{L}_t^{\frac{1}{2}}\right]\boldsymbol{L}_t^{-\frac{1}{2}}\boldsymbol{M}_{\boldsymbol{B}_t}\left(\boldsymbol{A}_t \boldsymbol{R}_t^{\frac{1}{2}} \boldsymbol{A}_t^\top\right)^{-1}$.
  $\boldsymbol{\Delta}_{\boldsymbol{A}_t} = \left(\boldsymbol{B}_t^\top \boldsymbol{L}_t^{\frac{1}{2}} \boldsymbol{B}_t\right)^{-1}\boldsymbol{M}_{\boldsymbol{A}_t}\boldsymbol{R}_t^{-\frac{1}{2}}\left[\boldsymbol{I} - \frac{1}{2}\boldsymbol{R}_t^{\frac{1}{2}}\boldsymbol{A}_t^\top\left(\boldsymbol{A}_t \boldsymbol{R}_t^{\frac{1}{2}} \boldsymbol{A}_t^\top\right)^{-1}\boldsymbol{A}_t\right]$.
  $\boldsymbol{B}_{t+1} = (1-\lambda\eta_t)\boldsymbol{B}_t - \eta_t \frac{\sqrt{1-\beta_1^t}}{1-\beta_3^t}\boldsymbol{\Delta}_{\boldsymbol{B}_t}$, $\boldsymbol{A}_{t+1} = (1-\lambda\eta_t)\boldsymbol{A}_t - \eta_t \frac{\sqrt{1-\beta_1^t}}{1-\beta_3^t}\boldsymbol{\Delta}_{\boldsymbol{A}_t}$.
end for
Note: factor gradients $\boldsymbol{G}_{\boldsymbol{B}_t} = \partial \mathcal{L}/\partial \boldsymbol{B}_t$ and $\boldsymbol{G}_{\boldsymbol{A}_t} = \partial \mathcal{L}/\partial \boldsymbol{A}_t$ are obtained from autograd. The weight-space gradient $\boldsymbol{G}_t$ can be obtained via a backward hook on $\boldsymbol{W}$. Add $\epsilon \boldsymbol{I}$ to matrix $\boldsymbol{B}_t^\top \boldsymbol{L}_t^{\frac{1}{2}} \boldsymbol{B}_t$ if it is not invertible.

We now analyze the computational complexity of computing the updates $\boldsymbol{\Delta}_{\boldsymbol{A}_t}$ and $\boldsymbol{\Delta}_{\boldsymbol{B}_t}$. For simplicity, we focus on $\boldsymbol{\Delta}_{\boldsymbol{A}_t}$, as the complexity for $\boldsymbol{\Delta}_{\boldsymbol{B}_t}$ is symmetric.

- Compute gradient $\boldsymbol{G}_t$. The stochastic gradient $\boldsymbol{G}_t$ of $\boldsymbol{W}_t$ is obtained during the backpropagation process.
- Row and column sums for $\boldsymbol{l}_t$ and $\boldsymbol{r}_t$. Compute $\boldsymbol{l}_t$ and $\boldsymbol{r}_t$ by summing the square of the element of $\boldsymbol{G}_t$ along rows or columns, which is in the computation $\mathcal{O}(mn)$. $\boldsymbol{L}_t^{\frac{1}{2}}$ and $\boldsymbol{L}_t^{-\frac{1}{2}}$ can be computed in $\mathcal{O}(m)$, $\boldsymbol{R}_t^{\frac{1}{2}}$ and $\boldsymbol{R}_t^{-\frac{1}{2}}$ can be computed in $\mathcal{O}(n)$.
- Compute $(\boldsymbol{B}_t^\top \boldsymbol{L}_t^{\frac{1}{2}} \boldsymbol{B}_t)^{-1}\boldsymbol{G}_{\boldsymbol{A}_t}\boldsymbol{R}_t^{-\frac{1}{2}}$. First to compute the inverse matrices $(\boldsymbol{B}_t^\top \boldsymbol{L}_t^{\frac{1}{2}} \boldsymbol{B}_t)^{-1}$ in $\mathcal{O}((m+r)r^2)$. Then multiply the inverse $(\boldsymbol{B}_t^\top \boldsymbol{L}_t^{\frac{1}{2}} \boldsymbol{B}_t)^{-1}$ by $\boldsymbol{G}_{\boldsymbol{A}_t}$ in $\mathcal{O}(nr^2)$, and multiply the diagonal matrix $\boldsymbol{R}_t^{-\frac{1}{2}}$ in $\mathcal{O}(nr)$.
- Compute $(\boldsymbol{B}_t^\top \boldsymbol{L}_t^{\frac{1}{2}} \boldsymbol{B}_t)^{-1}\boldsymbol{G}_{\boldsymbol{A}_t}\boldsymbol{A}_t^\top(\boldsymbol{A}_t \boldsymbol{R}_t^{\frac{1}{2}} \boldsymbol{A}_t^\top)^{-1}\boldsymbol{A}_t$. First to compute the inverse matrices $(\boldsymbol{A}_t \boldsymbol{R}_t^{\frac{1}{2}} \boldsymbol{A}_t^\top)^{-1}$ in $\mathcal{O}((n+r)r^2)$. Use the result from the last step, multiply $(\boldsymbol{B}_t^\top \boldsymbol{L}_t^{\frac{1}{2}} \boldsymbol{B}_t)^{-1}\boldsymbol{G}_{\boldsymbol{A}_t}$ by $\boldsymbol{A}_t^\top$ in computation $\mathcal{O}(nr^2)$, then multiply $(\boldsymbol{A}_t \boldsymbol{R}_t^{\frac{1}{2}} \boldsymbol{A}_t^\top)^{-1}$ in computation $\mathcal{O}(r^3)$, and multiply $\boldsymbol{A}_t$ in $\mathcal{O}(nr^2)$.

The computation complexity of $\mathbf{\Delta}_{\boldsymbol{A}_t}$ is $\mathcal{O}(mn+(m+n)r^2+r^3)$. The computation of $\mathbf{\Delta}_{\boldsymbol{B}_t}$ follows a similar structure, with symmetric terms. Its complexity is also $\mathcal{O}(mn+(m+n)r^2+r^3)$. Then we have

- **Per Iteration Computational Complexity.** Combining the computations of $\mathbf{\Delta}_{\boldsymbol{A}_t}$ and $\mathbf{\Delta}_{\boldsymbol{B}_t}$, the total computation complexity per iteration is $\mathcal{O}(mn+(m+n)r^2+r^3)$.

- **Memory Complexity.** The algorithm requires storing the vectors $\boldsymbol{l}_t$ and $\boldsymbol{r}_t$ in each iteration, hence the memory complexity is $\mathcal{O}(m+n)$.

F Supplementary Experiments of GPT-2 Fine-tuning

F.1 Parameter Settings

To ensure the reproducibility of the experiments described in Section 4 and to facilitate verification and comparison by others, we provide the complete details of the experimental parameter settings. Tables 8 and 9 list the parameters used during the fine-tuning of GPT-2 models and the learning rates corresponding to different optimizers, respectively. Specifically, we conduct experiments with GPT-2 models of various sizes. "Rank 4 (M)" represents a medium-sized model using LoRA with rank 4, while "Rank 4", "Rank 16", and "Rank 64" represent small models using LoRA with ranks 4, 16, and 64, respectively. To ensure the fairness of the experimental setup, we follow the parameter settings in LoRA [15] and Riemannian Preconditioned LoRA [38]. However, considering the sensitivity of different optimizers to learning rates, we use a grid search strategy to independently tune the optimal learning rate for each optimizer. This ensures that each optimizer operates under its best-performing configuration, providing more objective and reliable experimental results.

Table 8: Training and Inference Configuration for GPT-2 Fine-tuning.

Training		**LoRA** α		**Inference**	
Parameter	Value	Parameter	Value	Parameter	Value
Dropout Probability	0.1				
Batch Size	8				
Number of Epochs	5	α (for Rank 4)	32	Beam Size	10
Warm-up Steps	500	α (for Rank 16)	32	Length Penalty	0.8
Learning Rate Scheduler	Linear	α (for Rank 64)	128	No Repeat Ngram Size	4
Label Smoothing	0.1				
Weight Decay	0.01				

Table 9: Core Optimizer Parameters for GPT-2 fine-tuning.

Methods	Learning Rate ($\times 10^{-3}$)				β_3	$\beta_1=\beta_2$
	Rank 4	Rank 4 (M)	Rank 16	Rank 64		
SGD	90	90	200	90	/	/
Scaled GD	20	20	40	10	/	/
LoRA-Pro SGD	40	40	40	40	/	/
FILO-Adafactor	0.05	0.05	0.5	0.8	/	0.98
AdamW	0.2	0.2	0.2	0.2	0.9	0.999
Scaled AdamW	0.8	0.8	2	4	0.7	0.8
LoRA-Pro AdamW	0.1	0.1	0.2	0.4	0.9	0.999
FILO-AdafactorM	0.5	0.1	0.8	0.3	0.9	0.98

F.2 Cross-rank ablation on GPT-2 small

To complement the $r=4$ comparisons in the main text (Table 2), we report cross-rank scores at $r \in \{16, 64\}$ on GPT-2 small in Table 10. Hyperparameters follow Table 8 and Table 9. Across

both ranks and both optimizer families, FILO-Adafactor achieves the best or tied-best score on essentially every metric; gains over Scaled GD/AdamW and LoRA-Pro narrow at higher rank but remain positive, consistent with the main-text observation that the gradient-statistics-aware projection retains an advantage even when the LoRA factorization is less rank-constrained.

Table 10: GPT-2 small fine-tuned on E2E at $r \in \{16, 64\}$ (cross-rank ablation). Bold/underline = best/second-best per metric per (rank, optimizer family).

r	Method	E2E				
		BLEU	NIST	MET	ROUGE-L	CIDEr
16	SGD	65.4	8.07	40.7	67.0	2.07
	Scaled GD	<u>68.8</u>	<u>8.75</u>	45.0	69.2	<u>2.39</u>
	LoRA-Pro SGD	68.3	8.67	<u>45.1</u>	<u>69.3</u>	2.37
	FILO-Adafactor (ours)	**70.0**	**8.82**	**46.6**	**71.6**	**2.53**
	AdamW	69.5	8.77	46.4	71.2	2.48
	Scaled AdamW	<u>69.8</u>	<u>8.79</u>	46.5	<u>71.7</u>	<u>2.51</u>
	LoRA-Pro AdamW	69.7	8.73	**46.8**	<u>71.7</u>	<u>2.51</u>
	FILO-AdafactorM (ours)	**70.2**	**8.85**	<u>46.6</u>	**71.9**	**2.52**
64	SGD	64.7	8.08	40.8	66.7	2.04
	Scaled GD	68.5	8.68	45.0	69.4	<u>2.38</u>
	LoRA-Pro SGD	<u>68.6</u>	<u>8.71</u>	<u>45.4</u>	<u>69.7</u>	<u>2.38</u>
	FILO-Adafactor (ours)	**70.1**	**8.85**	**46.7**	**71.8**	**2.53**
	AdamW	69.6	8.76	<u>46.7</u>	<u>71.5</u>	2.50
	Scaled AdamW	<u>70.0</u>	<u>8.83</u>	46.4	<u>71.5</u>	2.50
	LoRA-Pro AdamW	<u>70.0</u>	8.82	46.6	<u>71.5</u>	<u>2.51</u>
	FILO-AdafactorM (ours)	**70.2**	**8.84**	**46.8**	**72.1**	**2.52**

F.3 Training Efficiency Comparison

To validate the training and inference efficiency of FILO-Adafactor, we report in Table 11 the total training and inference time required for all algorithms on the GPT-2 small model (rank 64) fine-tuned on E2E.

Table 11: Training and inference time (hours) of GPT-2 small (rank=64) fine-tuned with different optimizers on E2E.

Method	SGD	Scaled GD	LoRA-Pro SGD	FILO-Adafactor
Training	1.79	1.92	2.78	2.04
Inference	1.86	1.87	1.58	1.89
Method	AdamW	Scaled AdamW	LoRA-Pro AdamW	FILO-AdafactorM
Training	1.79	1.94	2.93	2.04
Inference	1.87	1.88	1.89	1.86

G Supplementary Experiments of Diffusion Model Fine-tuning

We evaluate FILO-Adafactor on diffusion-model personalization with the Mix-of-Show framework [10], which integrates Embedding Decomposed LoRA (EDLoRA) into the text encoder and U-Net of a Stable Diffusion backbone. Following [10, 38], we disable embedding-vector tuning and fine-tune only the LoRA components of the text encoder and U-Net submodules. The CLIP score [13] (ViT-B/32 variant of the CLIP model [28]) measures alignment between generated images and text prompts (range $[0, 100]$, higher better); FID [14] measures the distributional similarity between generated and reference images (lower better). Aggregate CLIP/FID scores at LoRA scaling factors $\{0.7, 1.0\}$ are in the main text (Table 6); this section reports per-prompt qualitative samples at

Table 12: Optimizer Parameters for fine-tuning the Mix-of-Show Model.

Methods	Learning Rate		β_3	$\beta_1 = \beta_2$
	Text-Encoder	U-Net		
SGD	1e-1	1e-1	/	/
Scaled GD	1e-1	1e-1	/	/
LoRA-Pro SGD	1e-1	1e-1	/	/
FILO-Adafactor	1e-5	1e-5	/	0.98
AdamW	1e-5	1e-4	0.9	0.999
Scaled AdamW	1e-5	1e-4	0.7	0.8
LoRA-Pro AdamW	1e-5	1e-5	0.9	0.999
FILO-AdafactorM	1e-5	1e-5	0.9	0.98

the same two scaling factors, using the per-optimizer learning rates listed in Table 12. At $s = 1.0$, Figures 3 and 4 compare AdamW-based optimizers on Harry Potter and Hermione Granger prompts respectively, and Figures 5 and 6 compare SGD-based optimizers on the same prompts. At $s = 0.7$, Figures 7 and 8 report the corresponding AdamW comparisons. Across both scaling factors, FILO-Adafactor preserves character identity and follows the prompt scene layout while remaining visually consistent across prompts.



Figure 3: Generated results based on the prompt “Harry Potter is walking near Mount Fuji” when fine-tuned using AdamW-based optimizers. All optimizers employed a LoRA scaling factor of 1.0, with the best learning rate. The results indicate that the output of the model trained with our optimizer incorporates the character “Harry Potter”, the action“walking”, and the scene “Mount Fuji”, yielding superior image quality compared to alternative approaches.



Figure 4: Generation results from the prompt "A photo of Hermione Granger on the beach, small waves, detailed symmetric face, beautiful composition" using AdamW-based optimizers. All the optimizers apply LoRA scaling factor as 1.0, with the best learning rate. Results demonstrate that the model trained with our optimizer generates higher-quality images than others, especially the face of Hermione Granger and the scene.



Figure 5: Generated results based on the prompt "Harry Potter standing near the lake" when fine-tuned using SGD-based optimizers. All optimizers employed a LoRA scaling factor of 1.0, with the best learning rate. Results demonstrate that the output images of the model trained with our optimizer have higher-quality than others, especially the face of Harry Potter.



Figure 6: Generated results based on the prompt "Hermione Granger wearing a brown shirt" when fine-tuned using SGD-based optimizers. All optimizers employed a LoRA scaling factor of 1.0, with the best learning rate. Results demonstrate that the model trained with FILO-Adafactor generates higher-quality images than others, especially the face of Hermione Granger.



Figure 7: Generated results based on the prompt "Harry Potter wearing a brown hat" when fine-tuned using AdamW-based optimizers. All optimizers employed a LoRA scaling factor of 0.7, with the best learning rate. The results indicate that the output of the model trained with FILO-Adafactor incorporates the character "Harry Potter", and the "hat", yielding superior quality compared to alternative approaches.



Figure 8: Generation results from the prompt "A photo of Hermione Granger on the beach, small waves, detailed symmetric face, beautiful composition" using AdamW-based optimizers. All the optimizers apply LoRA scaling factor as 0.7. Results demonstrate that FILO-Adafactor generates higher-quality images for both scaling factors than others, including the face of Hermione Granger and the scene.

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 abstract and introduction include the claims made in the paper.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: Discussed in Appendix.

   Guidelines:

   - The answer [N/A] 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: Detailed proofs are provided in Appendix.

Guidelines:

- The answer [N/A] 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: The hyperparameter settings are specified in Appendix.

   Guidelines:

   - The answer [N/A] 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: Code is submitted as Supplementary Material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: Experimental settings are provided in Appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: This paper reports mean scores.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: All the information is provided.

   Guidelines:

   - The answer [N/A] 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: This paper satisfies Code of Ethics.

   Guidelines:

   - The answer [N/A] 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: This paper discussed broader impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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: [N/A]

    Justification: This paper does not have such risk.

    Guidelines:

    - The answer [N/A] 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: The assets are cited and the license and terms are respected.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: This paper does not involve crowdsourcing nor search with human subjects.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: This paper does not involve crowdsourcing nor search with human subjects.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: This paper uses LLMs only for editing.

Guidelines:

- The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.