Accelerating LLM Pre-Training through Flat-Direction Dynamics Enhancement

Shuchen Zhu[*] Peking University
Rizhen Hu[*] Peking University
Mingze Wang[†] Peking University
Mou Sun Zhejiang Lab, Zhejiang
Xue Wang Meituan, Beijing
Kun Yuan[‡] Peking University
Zaiwen Wen Peking University

Abstract

Pre-training Large Language Models requires immense computational resources, making optimizer efficiency essential. The optimization landscape is highly anisotropic, with loss reduction driven predominantly by progress along flat directions. While matrix-based optimizers such as Muon and SOAP use fine-grained curvature information to outperform AdamW, their updates tend toward isotropy, which is relatively conservative along flat directions yet potentially aggressive along sharp ones. To address this limitation, we first develop a unified Riemannian Ordinary Differential Equation (ODE) framework that offers a continuous-time interpretation of adaptive algorithms: the preconditioner induces a Riemannian geometry that mitigates ill-conditioning, while momentum serves as a Riemannian damping term that promotes convergence. Guided by these insights, we propose **LITE**, a generalized acceleration strategy that enhances training dynamics by applying larger Hessian damping coefficients and learning rates along flat trajectories. Extensive experiments demonstrate that LITE significantly accelerates both Muon and SOAP across diverse architectures (Dense, MoE), parameter scales (130M to 1.3B), datasets (C4, Pile), and learning-rate schedules (cosine, wsd). Theoretical analysis confirms that LITE facilitates faster convergence along flat directions in anisotropic landscapes, providing a principled approach to efficient LLM pre-training. The code is available at https://github.com/SHUCHENZHU/LITE.

1 Introduction

Large language models (LLMs) have advanced artificial intelligence with exceptional capabilities, yet the pre-training procedure is computationally intensive due to massive model and data scales. Improving pre-training efficiency is essential for continued scaling, and the optimizer choice is crucial. Currently, AdamW [17, 24] serves as the standard in most LLM pre-training pipelines, favored for its simplicity and efficiency. However, its coordinate-wise scaling (diagonal preconditioner) ignores off-diagonal Hessian correlations, limiting the ability to navigate complex optimization landscapes.

To address the limitation of the diagonal preconditioning, algorithms like Shampoo [11, 33] and KFAC [27] introduce matrix-based preconditioners to capture richer geometric structures, demonstrating superior convergence over AdamW in traditional deep learning tasks. Advancing this paradigm to the scale of LLM pre-training, a new wave of matrix-based optimizers, such as Muon [16] and SOAP [43], has emerged, achieving significant performance gains over AdamW. Notably, Muon has already

[*]Equal contribution.
[†]Project lead.
[‡]Correspondence to: Kun Yuan <kunyuan@pku.edu.cn>

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

achieved validated success in industrial-scale training scenarios [22, 40]. These methods signify a paradigm shift towards exploiting matrix-level curvature information to accelerate convergence.

Despite the empirical success of these advanced optimizers, several limitations remain:

1. **Preconditioner-induced tendency toward isotropic update.** Although adaptive methods correct descent directions by effective preconditioners, prior studies [36, 60, 22, 25, 47] suggest that the *update magnitudes* often tend towards *isotropic*, i.e., of comparable scale across different Hessian eigen-directions (refer to more discussions in Appendix D.1). This behavior is suboptimal in the ill-conditioned landscape, which is too cautious along flat directions while potentially aggressive along sharp ones.
2. **Inadequate momentum mechanisms for non-convexity.** Limited by the simple linear nature of Exponential Moving Average (EMA) that inherently imposes an isotropic damping effect, existing momentum schemes inadequately exploit second-order information, leaving them susceptible to ill-conditioned curvature. While techniques like Nesterov acceleration implicitly incorporate Hessian-driven damping [32], the design largely stems from convex optimization paradigms, making them insufficient for effectively accelerating optimization along non-convex directions.

Addressing these limitations requires understanding how preconditioning and different momentum mechanisms jointly influence training dynamics in anisotropic landscapes. This motivates the following questions:

- How to establish a principled theoretical understanding of the joint roles of the momentum and preconditioner in adaptive optimization algorithms?
- Guided by this foundation, how can we rectify the inadequate isotropic update and refine the momentum mechanism to achieve superior efficiency in pre-training LLMs?

Our contributions are as follows:

- We develop a **unified Riemannian ODE framework** that encapsulates prevalent adaptive optimizers for LLM pre-training (*e*.g. AdamW, Muon, SOAP) and their Nesterov-accelerated counterparts. This framework provides a continuous-time interpretation the joint mechanism between preconditioning and momentum from a continuous time perspective: the preconditioner induces a Riemannian geometry that mitigates the landscape's ill-condition, while momentum functions as a *Riemannian damping* to foster optimization within this metric.
- Building on this framework, we propose **LITE**, a lightweight generalized strategy for accelerating adaptive optimizers in LLM pre-training. Using our ODE analysis, LITE applies **larger Hessian damping coefficients and learning rates in the flat directions**, which enhances momentum accumulation and amplifies update magnitudes therein. This approach is designed to *accelerate training dynamics along the flat directions* that dominate loss reduction within the ill-conditioned landscape, thereby enhancing the pre-training performance. Furthermore, we provide a **theoretical analysis** of the training dynamics of LITE within anisotropic landscapes, showing that it boosts dynamics in flat directions and facilitates optimization, thereby supporting our design intuition.
- We conduct **extensive empirical evaluations** on the acceleration performance of LITE across diverse LLM pre-training settings, spanning different base optimizers (Muon, SOAP), model architectures (LLaMA, QwenMoE), datasets (C4, Pile), and learning rate schedules (cosine, wsd), with model sizes from 0.13B to 1.3B parameters. Our results demonstrate that LITE-accelerated optimizers achieve remarkable loss reductions and more favorable scaling laws compared to standard baselines, delivering notable speedups in long-horizon training (Figure 1). This indicates the potential of LITE for superior scalability to larger models and extended token budgets.

2 Related Work

Accelerated Optimizers for LLM pre-training. Adaptive optimizers serve as the cornerstone of modern LLM pre-training. One stream of research focuses on refining the preconditioner: moving beyond simple element-wise scaling of AdamW [24], methods like [21, 44] incorporate explicit curvature estimation, while matrix-based approaches [16, 43, 30, 23, 18, 34] use richer geometric structures to better approximate the curvature. Complementing these geometric improvements are

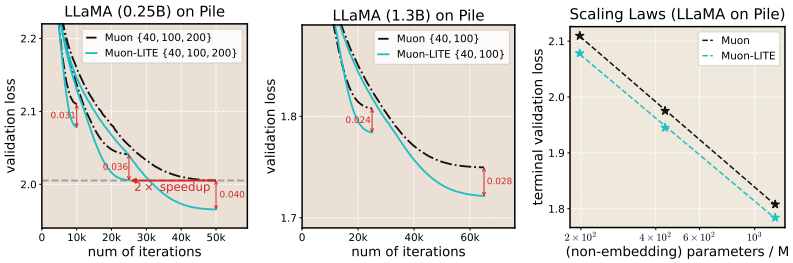


Figure 1: Muon-LITE exhibits superior scaling behavior across varying token budgets (left, middle) and model sizes (right) compared to Muon, showing a sustained, non-decaying advantage with increased training. Legend numbers denote the token-to-parameter ratio for each curve (e.g., $100\times$).

refinements to the momentum mechanism, spanning multi-timescale momentum [29], Nesterov-type acceleration [52, 58], and various momentum correction schemes [15, 20].

Training Dynamics and Loss Landscapes in Deep Learning. Prior studies [10, 56, 59, 38] demonstrate the *ill-conditioned and anisotropic* nature of the deep learning loss landscapes. Specifically, the Hessian spectrum is dominated by a massive bulk of near-zero and negative eigenvalues (referred to as *flat directions*), while the large positive eigenvalues are significantly greater in magnitude but sparse in number (referred to as *sharp directions*), as schematically illustrated in Figure 2 (left). Recent investigations [35, 5, 48, 44] into the training dynamics of gradient-based algorithms on this landscape have led to the consensus that the anisotropic landscape induces a distinct *time-scale separation* in the training dynamics (Figure 2 (right)): **1) Fast dynamics:** Along *sharp* directions, the dynamics exhibit rapid but non-divergent oscillations, which dictate training stability yet contribute minimally to loss reduction. **2) Slow dynamics:** Along *flat* directions, the dynamics evolves steadily but slowly, dominating total loss reduction.

Informed by these insights, we propose LITE to accelerate training by boosting the slow training dynamics along flat directions (Figure 2 (right)). More related works are in Appendix A.

3 Preliminary

Notations. Let v_k denote the k-th component of a vector v. For a positive definite matrix F, we define the induced inner product and norm as $\langle u, v\rangle_F = u^\top F v$ and $\|u\|_F = \sqrt{u^\top F u}$ respectively. We distinguish between Riemannian operators $(\mathrm{grad}, \mathrm{Hess})$ and their Euclidean counterparts (∇, ∇^2). The symbol $\odot$ represents the element-wise product, and $\otimes$ denotes the Kronecker product. A subscript or superscript F is occasionally attached to geometric operators (*e.*g. ∇^F) to emphasize they are induced by the Riemannian metric F.

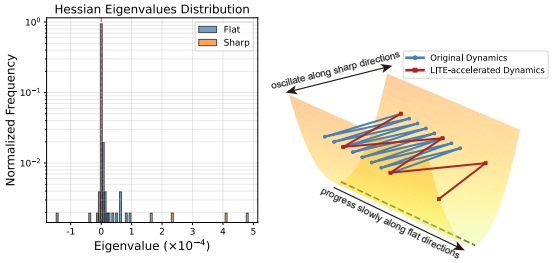


Figure 2: (left) Hessian eigenvalue distribution of an `up_proj` FFN block in a toy LLaMA model; (right) Schematic illustration of the time-scale separation property and the acceleration mechanism of LITE.

3.1 Manifold Optimization

Manifold optimization deals with variables constrained to a Riemannian manifold. In this work, we consider a special case: the variable (parameter) space $\mathbb{R}^p$ not as a standard Euclidean space, but as a Riemannian manifold $\mathcal{M} = (\mathbb{R}^p, F)$ equipped with metric $F(w) \in \mathbb{R}^{p\times p}$ at each $w \in \mathbb{R}^p$. This geometric perspective allows us to analyze the optimization landscape through the lens of local curvature defined by F. Detailed discussions are provided in Appendix F.

Tangent and Cotangent Spaces. At any point $w \in \mathcal{M}$, the tangent space $T_w\mathcal{M} \cong \mathbb{R}^p$ represents the vector space of all possible directional velocities. The Riemannian metric induces a norm $\|\cdot\|_{F(w)}$

on the tangent space. The cotangent space $T_w^*\mathcal{M}$ is defined as the dual space to $T_w\mathcal{M}$, which is equipped with the norm $\|\cdot\|_{F(w)^{-1}}$ and is also isomorphic to $\mathbb{R}^p$. The metric induces linear isomorphisms $F(w)^{-1}: T_w^*\mathcal{M} \to T_w\mathcal{M}$ and $F(w): T_w\mathcal{M} \to T_w^*\mathcal{M}$. While the Euclidean gradient $\nabla f(w)$ naturally resides in $T_w^*\mathcal{M}$, the Riemannian gradient is obtained by mapping it back to the tangent space via $\operatorname{grad} f(w) = F(w)^{-1}\nabla f(w)$. It gives the steepest descent steps in the local geometry by $-\frac{F(w)^{-1}\nabla f(w)}{\|F(w)^{-1}\nabla f(w)\|_{F(w)}} = \lim_{\epsilon\to 0}\frac{1}{\epsilon}\operatorname{argmin}_{\|h\|_{F(w)}\le\epsilon}(f(w+h)-f(w))$.

Levi-Civita Connection. Levi-Civita connection $\nabla_{(\cdot)}(\cdot)$ provides a rigorous tool for differentiating vector fields on Riemannian manifolds, generalizing the concept of Euclidean directional derivatives to curved geometries. For vector fields u, v, $\nabla_u v$ yields a vector field that quantifies how v twists or deviates from a parallel field as it moves along u. See more discussions in Appendix F.2.

3.2 ODE Perspective of Momentum-based Algorithms

Continuous-time modeling via ODEs constitutes a rigorous and principled paradigm for analyzing momentum-based algorithms, offering *deep theoretical insights and clear intuitions* into their underlying dynamics. [39, 32, 3] introduced an inertial system with Hessian damping (ISHD) that can depict the momentum-based algorithms in higher resolution, which takes the form

$$\ddot{w}_t + \alpha_t\dot{w}_t + \beta_t\nabla^2 f(w_t)\dot{w}_t + \gamma_t\nabla f(w_t) = 0, \tag{1}$$

where $\alpha_t, \beta_t, \gamma_t \ge 0$ denote the coefficients for momentum decay, Hessian damping, and the gradient driving force, respectively. Discretizing (1) recovers various momentum methods: specifically, setting $\beta_t = 0$ yields Heavy Ball momentum, while $\beta_t > 0$ corresponds to Nesterov-type momentum. Crucially, the Hessian damping term $\nabla^2 f(w_t)\dot{w}_t$ and its discretized counterpart $\nabla f(w_k) - \nabla f(w_{k-1})$ serve to mitigate oscillations, enhancing the optimization stability. Further details on discretization, equivalent formulations, and the role of Hessian damping are provided in Section E.

4 A Unified Riemannian ODE Framework for Understanding Adaptive Optimizers

Adaptive optimizers combine momentum with preconditioners, using momentum-induced acceleration while simultaneously adapting step sizes anisotropically for efficient training. In this section, we interpret their mechanics by modeling these optimizers as inertial ODE systems with Hessian damping on Riemannian manifolds (RISHD) equipped with preconditioner-induced metrics.

4.1 A Unified Discrete Formulation

Formally, the update rules of most adaptive optimizers can be encapsulated in a unified formulation:

$$\begin{cases} m_k = (1-\alpha)m_{k-1} + \nabla f(w_k), \\ w_{k+1} = w_k - \eta_k F(w_k)^{-1}(m_k + \beta\nabla f(w_k)), \end{cases} \tag{2}$$

where w_k denotes the parameters, m_k is the momentum [4], $F(w_k)^{-1}$ serves as the preconditioner, and the term $\beta\nabla f(w_k)$ yields a Nesterov-type momentum. In neural network training, the design of the preconditioner F^{-1} typically involves three features: *1) Block-Diagonal Structure*, treating F block-wise to reduce computational overhead; *2) EMA Estimation*, smoothing mini-batch noise to stabilize curvature estimation; and *3) Hessian Alignment*, where F approximates the Hessian (*e*.g. diagonal Hessian) or its structurally aligned proxy (*e*.g. uncentered gradient covariance $(\mathbb{E}[gg^\top])^{1/2}$ and gradient whitening matrix, related to the Gauss-Newton part of Hessian). By varying the structure of F and the momentum coefficient β, **the formulation (2) subsumes a broad family of adaptive methods**, including N-AdamW, Muon, and SOAP. See Table 1 and Appendix I for further details.

Optimizers using Hessian Damping Several recent optimizers (*e*.g. Lion, MARS [58], and Muon) incorporate Nesterov-style momentum to surpass the performance of vanilla Heavy Ball momentum. The key step is preconditioning $m_k + \beta g_k$ as in (2) rather than the standard momentum m_k. The βg_k term induces an implicit Hessian damping effect by $\nabla f(w_k) - \nabla f(w_{k-1}) \approx \nabla^2 f(w_k)(w_k - w_{k-1})$, crucial for reducing oscillations in ill-conditioned landscapes (Appendix E).

[4] The missing coefficient α of ∇f can be absorbed in η_k.

Table 1: Update forms (matrix and vector) and their estimated deterministic proxies of the metric F for common adaptive optimizers. m denotes the EMA-type momentum, g represents the stochastic gradient, and $M_\beta = M + \beta G$. Capital letters denote the matrix forms corresponding to their vector counterparts (lowercase). For AdamW, we present its Nesterov-accelerated variant N-AdamW. See Appendix I for further discussions and details.

Algorithms	Matrix Form	Vector Form	F (up to a constant factor)
N-AdamW	$\frac{M+\beta G}{\sqrt{V}}$	$\operatorname{diag} v^{-\frac{1}{2}}(m+\beta g)$	$(\operatorname{diag}\mathbb{E}[gg^\top])^{\frac{1}{2}}$
Muon	$M_\beta(M_\beta^\top M_\beta)^{-\frac{1}{2}}$	$((M_\beta^\top M_\beta)\otimes I)^{-\frac{1}{2}}(m+\beta g)$	$(\mathbb{E}[G^\top G])^{\frac{1}{2}}\otimes I$
SOAP	$Q_l\frac{Q_l^\top M Q_r}{\sqrt{V_{\text{rot}}}}Q_r^\top$	$(Q_r\otimes Q_l)(\operatorname{diag} v_{\text{rot}})^{-\frac{1}{2}}(Q_r^\top\otimes Q_l^\top)m$	$(Q_r\otimes Q_l)(\operatorname{diag}\mathbb{E}[g_{\text{rot}}g_{\text{rot}}^\top])^{\frac{1}{2}}(Q_r^\top\otimes Q_l^\top)$

4.2 The Riemannian ODE Framework

Intuition from the discrete update. For brevity, we first assume that $F(w)\equiv F$ and $\eta_k\equiv\eta$ are constant. Then eliminating the momentum variable m_k from (2) yields

$$w_{k+1} = w_k + (1-\alpha)(w_k - w_{k-1}) - \eta(1+\alpha\beta)F^{-1}\nabla f(w_k) \\ - \eta\beta(1-\alpha)F^{-1}\big(\nabla f(w_k) - \nabla f(w_{k-1})\big).$$

Using the Taylor approximation $\nabla f(w_k) - \nabla f(w_{k-1}) \approx \nabla^2 f(w_k)(w_k - w_{k-1})$, this scheme corresponds to the continuous-time limit

$$\ddot{w}_t + \alpha\dot{w}_t + \eta\beta(1-\alpha)F^{-1}\nabla^2 f(w_t)\dot{w}_t + \eta(1+\alpha\beta)F^{-1}\nabla f(w_t) = 0.$$

In this form, the Heavy-Ball component $\alpha\dot{w}_t$ induces isotropic damping, whereas the Nesterov-type momentum $F^{-1}\nabla^2 f(w_t)\dot{w}_t$ contributes additional Hessian damping under the F-induced geometry. This intuition remains valid when $F(w)$ and η_k vary slowly. Next, we provide a quantitative continuous-time formulation to establish this (with a slight change of notations for brevity).

Continuous-time Viewpoint. The generalized form (2) can be interpreted as a approximated discretization of a Riemannian ODE. Using Levi-Civita connection $\nabla^F_{(\cdot)}(\cdot)$ to generalize the Euclidean derivatives to the covariant derivatives, we can extend the Euclidean ISHD (1) to the Riemannian ISHD (RISHD) on $(\mathcal{M}, F)$:

$$\nabla^F_{\dot{w}_t}\dot{w}_t + \alpha_t\dot{w}_t + \beta_t \operatorname{Hess}_F(w_t)\dot{w}_t + \gamma_t \operatorname{grad}_F f(w_t) = 0, \tag{3}$$

where Hess_F and grad_F denote F-induced Riemannian Hessian and gradient respectively.

First-order Tangent-Cotangent Formulation. We can introduce the momentum variable m_t tracking the velocity $\dot{w}_t$ to decouple (3) into a first order system that evolves in the tangent space (T) and cotangent space (C) respectively:

Proposition 1 (Adapted from Proposition 2). *If $\alpha_t = \alpha - \dot{\eta}_t/\eta_t$, $\beta_t = \beta\eta_t$, $\gamma_t = \eta_t(\alpha\beta+1)$, (3) is equivalent to the following ODE system (with a Riemannian correction term R_t defined in (20)):*

$$\begin{cases} \dot{w}_t = -\eta_t F(w_t)^{-1}(m_t + \beta\nabla f(w_t)), & \text{(T)} \\ \dot{m}_t = -\alpha m_t + \nabla f(w_t) + R_t. & \text{(C)} \end{cases}$$

The term R_t mainly depends on η_t and $\nabla F(w_t)$, and every summand of R_t carries a factor of η_t together with a derivative of F (see (20)). Thus R_t is typically negligible due to the small step size η_t and the stable preconditioner (implying a bounded $\nabla F(w_t)$). Thus we can ignore R_t for brevity. Beyond a qualitative interpretation, the exact equivalence in Proposition 1 quantitatively characterizes how the variation of F affects the curvature-induced error.

The decoupled formulation in (T) and (C) reveals the structure of RISHD in (3): Momentum evolves in the cotangent space, while the preconditioner F^{-1} maps it to the tangent space to drive the parameter update. This form also naturally gives the specific form in which modern adaptive methods incorporate momentum: the preconditioner is applied to the momentum accumulated from raw gradients, rather than accumulating preconditioned gradients by EMA. Furthermore, this Riemannian ODE framework is readily extensible to higher-order cases, enabling the characterization of complex momentum schemes such as the multi-timescale momentum in AdEMAMix [29] (Appendix E.3).

Remark 1. *A line of work has leveraged ODE techniques to characterize the continuous-time limit of accelerated gradient methods in non-Euclidean geometry [49, 6]. Compared to these works, our RISHD additionally provides a high-resolution interpretation tailored to Nesterov-type momentum under the Riemannian metric.*

Joint effect of the preconditioner and momentum. The continuous-time ODE framework (3) is instrumental and intuitive for understanding the joint mechanism of preconditioning and momentum, which remains opaque in discrete formulations. Assuming that $F(w)$ varies slowly and aligns well with the Hessian, the induced Riemannian metric effectively mitigates ill-conditioning, as it yields a Riemannian Hessian $\mathrm{Hess}_F\, f(w) \approx F^{-1}(w)\nabla^2 f(w)$ with much smaller condition number than the Euclidean Hessian $\nabla^2 f(w)$. Within this improved geometry, the Riemannian damping $-(\alpha_t + \beta_t \,\mathrm{Hess}\, f(w))\dot{w}_t$ acts as the momentum effect, further accelerating optimization.

Connection with the Discrete Formulation. Compared to the original second-order system (3), the equivalent first-order coupled system offers a more direct path for numerical implementation. Applying a semi-implicit Euler discretization scheme to (T) and (C) with step size $h = 1$ and $R_t \approx 0$ yields the discrete formulation (2).

5 The LITE Approach

In this section, we introduce LITE, an acceleration strategy grounded in the characteristics of the loss landscape and the theoretical insights regarding the combined role of momentum and preconditioning established in Section 4.2. As discussed in Section 2, the trajectory along sharp directions exhibits significant fluctuations, and the efficiency of traversal along flat directions becomes the critical determinant of the final loss. Motivated by this, our strategy aims to **selectively accelerate optimization along flat directions while preserving stability in sharp directions.** To implement this, we adopt a decoupled tuning scheme: we anchor the hyper-parameters governing the sharp directions in (2) to maintain stability, while adaptively adjusting those corresponding to the flat directions. In the following discussion, we derive a heuristic adjustment scheme for the flat-direction hyper-parameters from the perspective of continuous-time dynamics in Section 4.

5.1 Acceleration Methods from the ODE Perspective

We begin by analyzing the role of momentum through the lens of the continuous-time limit in (1), focusing initially on the Euclidean case of (3) for clarity. Intuitively, the system models the dynamics of a particle moving within a potential field defined by f. The particle at w_t is subject to the combined effects of a driving force $-\gamma_t \nabla f(w_t)$ and a curvature-adaptive damping (drag force) $(\alpha_t I + \beta_t \nabla^2 f(w_t))(-\dot{w}_t)$ opposing its velocity $\dot{w}_t$, with its acceleration $\ddot{w}_t$ governed by Newton's Second Law. Along flat directions, we can enhance the accumulation of velocity by increasing the driving force coefficient γ_t while minimizing the effective damping coefficient $\alpha_t I + \beta_t \nabla^2 f(w_t)$ that is positive definite to avoid divergence. Since flat directions are often characterized by non-convexity, our strategy is **increasing γ_t and β_t in the flat directions.** [5]

The above discussion, framed in terms of damping and motion in Euclidean space, can be naturally extended to Riemannian manifolds (curved spaces). The corresponding physical background shifts from Newtonian mechanics to Lagrangian mechanics [2], yet the same conclusions hold. For sharp directions, the coefficients remain unchanged to maintain stability conditions. Given the relationships $\gamma_t = (1 + \alpha\beta)\eta_t$ and $\beta_t = \beta\eta_t$ in (3), we can amplify γ_t and β_t by increasing β and η_t. Finally, assuming well-aligned top eigenspaces between F and Hessian, the accelerated dynamics is given by:

$$\begin{cases} \dot{w}_t = -\eta_t F(w_t)^{-1} P_t(m_t + \beta_1 \nabla f(w_t)) - \chi \eta_t F(w_t)^{-1} Q_t(m_t + \beta_2 \nabla f(w_t)), \\ \dot{m}_t = -\alpha m_t + \nabla f(w_t), \end{cases} \tag{4}$$

where P_t is the projection to the sharp subspace, $Q_t = I - P_t$, and $\chi \geq 1$, $\beta_2 \geq \beta_1 \geq 0$ are hyper-parameters. Discretizing (4) like (2) yields the accelerated algorithms framework LITE (Algorithm 1). This formulation naturally yields amplified update magnitudes and enhanced momentum along

[5] Regarding damping, we do not tune α_t for simplicity. While α_t offers a similar mechanism for reducing damping, β_t is preferred as it provides curvature adaptivity.

flat directions, facilitating faster traversal through these slow-progressing regimes. An illustrative theoretical analysis of acceleration mechanism in LITE on a quadratic function is in Appendix H.

Algorithm 1 LITE Strategy

Input: $w_0, m_0, \{\eta_k\}, \alpha, \chi \geq 1, \beta_2 \geq \beta_1 \geq 0$.
for $k = 0$ **to** K **do**
  Update momentum $m_k = (1-\alpha)m_{k-1} + \nabla f(w_k)$.
  Estimate projection Q_k onto the flat directions. Then estimate the update direction

$$u_k = (I - Q_k)F(w_k)^{-1}(m_k + \beta_1 \nabla f(w_k)) + \chi Q_k F(w_k)^{-1}(m_k + \beta_2 \nabla f(w_k)).$$

  Update parameters $w_{k+1} = w_k - \eta_k u_k$.
end for

Remark 2. *Introducing χ is mathematically equivalent to modifying the preconditioner $F(w_k)^{-1}$ to $(I - Q_k)F(w_k)^{-1} + \chi Q_k F(w_k)^{-1}$. However, the inclusion of β_2 introduces a distinct adjustment on momentum damping along flat directions, which cannot be achieved solely by altering F^{-1}.*

5.2 Practical Implementation of LITE

In this section, we present practical methods for implementing LITE (Algorithm 1) in LLM-pretraining. We specifically target the anisotropic landscape inherent to each parameter block, aiming to efficiently identify its sharp directions (and flat directions via the orthogonal complement).

Efficiently approximating the flat directions Prior studies [26, 28] demonstrate that the Shampoo-class preconditioners effectively approximates the Gauss-Newton component, the dominant term in the Hessian. Since both Muon and SOAP adopt Shampoo-style Kronecker factorization to approximate the whitening metric, we hypothesize that *the top eigenspaces of their preconditioners and the Hessian are highly aligned.*

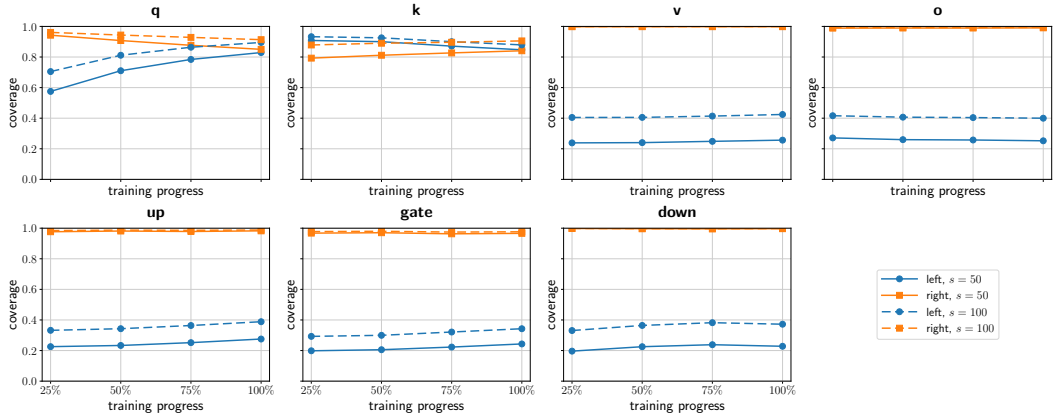


Figure 3: Evolution of the coverage score (defined in (8)) of each block's top-k Hessian eigenspace by the left and right top-s gradient singular subspaces in layer 2 of a 120M LLaMA model ($k = 200$, $s \in \{50, 100\}$). The model is trained for 20,000 steps on approximately 10B tokens. Scores closer to 1 indicate greater containment.

To verify this hypothesis, we conduct experiments on a 120M LLaMA model with $d_{\text{model}} = 768$ (see Appendix C.1 for details). For each weight matrix $W \in \mathbb{R}^{m \times n}$ in the transformer layers (q, k, v, o, up, gate, down), we assess whether the top-k subspace of the Hessian $H_W \in \mathbb{R}^{mn \times mn}$ can be well covered by the subspaces spanned by the top-s eigenspaces of $GG^\top$ and $G^\top G$ (with bases denoted by B_{left} and B_{right}, respectively), where G is the stochastic gradient matrix. The two subspaces have the following bases:

$$B_{\text{left}} = U_s \otimes I_n \in \mathbb{R}^{mn \times sn}, \qquad B_{\text{right}} = I_m \otimes V_s \in \mathbb{R}^{mn \times ms},$$

where $U_s \in \mathbb{R}^{m\times s}$ and $V_s \in \mathbb{R}^{n\times s}$ are the top-s eigenvectors of $GG^\top$ and $G^\top G$, respectively [6].

Figure 3 shows that throughout the whole training process, **the top eigenspaces of the block-wise Hessian can be well covered by the subspace spanned by B_{right}** (corresponding to the right singular space of G). The top-s subspace of $GG^\top$ (i.e., the left singular space) also shows good coverage in some blocks, but to a lesser degree than the right subspace. Since the preconditioners used in Muon and SOAP are closely tied to $G^\top G$ and $GG^\top$, we propose the following tailored implementation strategies.

Accelerating Muon (Muon-LITE) At iteration k, let $G_k \in \mathbb{R}^{m\times n}$ ($m \geq n$) represent the stochastic gradient for a given parameter block. Denote the momentum update $M_k = (1-\alpha)M_{k-1} + G_k$ and the Nesterov correction $\widetilde{M}_k = M_k + G_k/(1-\alpha)$ [7]. Recognizing that $\widetilde{M}_k^\top \widetilde{M}_k$ acts as a stabilized proxy for $G_k^\top G_k$, we utilize it to characterize the sharp directions according the previous discussion. Specifically, we construct the projection matrix $P_k \in \mathbb{R}^{n\times n}$ onto the top-d_s eigenspace of $\widetilde{M}_k^\top \widetilde{M}_k$, where $d_s n$ specifies the dimension of the sharp subspace. A numerically efficient, optimizer state-free method for estimating P_k is detailed in Appendix B.1. Consequently, the update direction is formulated as:

$$U_k = \widetilde{M}_k(\widetilde{M}_k^\top \widetilde{M}_k)^{-\frac{1}{2}}(P_k + \chi(I_n - P_k)) + G_k(G_k^\top G_k)^{-\frac{1}{2}}(\beta_1 P_k + \chi\beta_2(I_n - P_k)),$$

where $\beta_2 \geq \beta_1$ are the Hessian damping coefficients, and $\chi \geq 1$ denotes the amplification ratio of the learning rate in flat directions relative to sharp ones. Setting $\chi = 1$ and $\beta_{1,2} = 0$ recovers the original Muon baseline. Here we use the precondition $(G_k^\top G_k)^{-1/2}$ instead of $(\widetilde{M}_k^\top \widetilde{M}_k)^{-1/2}$ to the gradient term, since it has superior numerical stability and efficiency of applying NS iterations, and $G_k(G_k^\top G_k)^{-1/2}$ serves as a valid instantaneous estimator of gradient whitening (orthogonalization). In practice, χ typically requires eventual decay to realize its acceleration effect during training (see details in Appendix B.3).

Accelerating SOAP (SOAP-LITE) We retain the notations established above. The SOAP preconditioner explicitly provides the eigenvalues (V) and eigenvectors ($Q_r \otimes Q_l$) of the curvature estimation (Table 1). Thus we directly identify the sharp directions in the ($Q_r \otimes Q_l$)-rotated eigenspace by selecting the top-d_s ($d_s \leq mn$) elements of V. Let $P_k \in \mathbb{R}^{m\times n}$ denote the mask matrix indicating the sharp directions at iteration k, and let $Q_k = 1_m 1_n^\top - P_k$ represent the flat directions. The update direction for accelerating SOAP is given by:

$$U_k = Q_l\left[V_k^{-\frac{1}{2}} \odot (\beta_1 P_k + \chi\beta_2 Q_k) \odot (Q_l^\top G_k Q_r) + V_k^{-\frac{1}{2}} \odot (P_k + \chi Q_k) \odot (Q_l^\top M_k Q_r)\right] Q_r^\top,$$

where, similar to Muon-LITE, the SOAP baseline is recovered when $\chi = 1$ and $\beta_{1,2} = 0$. In practice, to enhance training stability, we implement P_k using a soft transition scheme (smoothly changing from 0 to 1) rather than a hard binary cutoff. See Appendix B.2 for implementation details.

6 Experiments

We evaluate LITE in accelerating Muon and SOAP on a wide range of LLM pre-training scenarios. Experimental details are provided in Appendices B and C.

6.1 Results on Dense Models

Main Results. Figures 4 and 10 compare the performance of Muon-LITE and SOAP-LITE against vanilla Muon and SOAP across a range of LLaMA model sizes on both the C4 and Pile datasets. The values of $\{\chi, \beta_{1,2}\}$ in LITE were determined via a search on the 0.25B model experiments and were then **used uniformly for other experiments.** Across all experimental settings, Muon-LITE and SOAP-LITE consistently achieve lower terminal losses than their respective well-tuned baselines. Besides, LITE outperforms both LITE-L and LITE-H, validating the effectiveness of simultaneously increasing χ and β_2. We visualize the scaling laws of Muon-LITE versus Muon in Figure 1 (right).

[6] Here we use row-major vectorization.

[7] We adopt the Nesterov formulation consistent with [16].

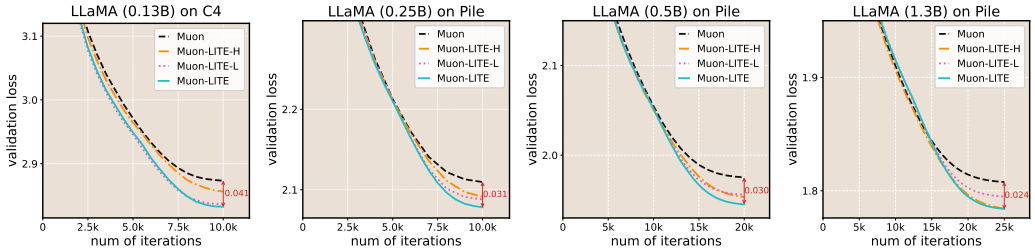


Figure 4: Performance comparison of Muon and Muon-LITE on LLaMA2 pre-training tasks with a `cos` learning rate schedule. The suffixes L and H denote two ablation variants of LITE: in flat directions, L increases only the learning rate ratio $\chi \geq 1$, while H increases only the Hessian damping coefficient β_2. The token batch size is approximately 2M for the 1.3B model, and 1M for others.

The performance gains yielded by Muon-LITE remain consistent across a wide range of model scales, indicating its potential scalability. Further downstream analyses are provided in Appendix D.

We subsequently investigate the acceleration efficacy of LITE over extended training horizons, increasing the training token budget from $40\times$ the parameter count to $100\times$ and $200\times$. Figure 1 (left, middle) indicates that the loss reduction gains yielded by Muon-LITE persist across these extended regimes, maintaining a significant speedup over Muon in longer training horizons. This demonstrates the sustained scalability of Muon-LITE with respect to the number of training tokens.

Ablation Studies and Hyper-parameter Robustness. We demonstrate that the acceleration strategy for flat directions is detrimental when applied to sharp directions, and that LITE is generally robust to hyper-parameter tuning, yielding notable acceleration across a wide range of configurations $(\chi, \beta_{1,2}, d_s)$. See Appendix D.2 for more details.

6.2 Results on MoE Models

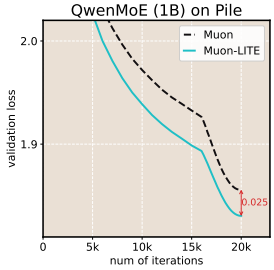


Figure 5: Results in QwenMoE pre-training tasks.

Mixture-of-experts (MoE) architectures have emerged as a critical paradigm for modern LLMs. By selectively activating a subset of experts for each token, MoE models achieve superior scaling efficiency over their dense counterparts.

We evaluate Muon-LITE against the Muon baseline on a 1B-parameter QwenMoE model, using a `wsd` learning rate schedule. Figure 5 demonstrates that Muon-LITE consistently outperforms Muon throughout the training process, ultimately achieving a significantly lower terminal loss. Furthermore, the performance gap between Muon-LITE and Muon progressively widens as iterations proceed during the stable phase. This trend suggests that the efficiency gains offered by Muon-LITE are likely to persist or even amplify at larger data scales.

7 Theoretical Analysis of Training Dynamics

In this section, we provide a theoretical characterization of the acceleration mechanism of LITE. We adopt and extend the *River-Valley* landscape framework in [48], which models the pre-training loss landscape as a deep valley with a flat river at its bottom. We *generalize this framework* to high-dimensional settings, and *analyze the dynamics* (4) corresponding to LITE (Algorithm 1). Our analysis demonstrates that LITE effectively accelerates the training dynamics along the River, which dictates the terminal loss. Below, we introduce the assumptions for our theoretical derivation.

Assumption 1 (River Structure)**.** *Fix an integer $d_s \in \mathbb{Z}_+$ and an open set $U \subset \mathbb{R}^p$. Let $P_s(w)$ denote the Euclidean projection onto the top d_s eigenspace of the Hessian $\nabla^2 f(w)$. We assume that $P_s(w)$ varies smoothly with respect to w, and that the set (referred to as* River*) $\mathcal{R} = \{w \in U : P_s(w)\nabla f(w) = 0\}$ constitutes a $(p - d_s)$-dimensional manifold. Furthermore, we assume that within the sharp subspace Range$(P_s(w))$, the Hessian $\nabla^2 f(w)$ and the preconditioner $F(w)$ share a common eigen-basis. Finally, the ODE $\frac{d}{dt}\psi_t(w) = -P_s(\psi_t(w))F^{-1}(\psi_t(w))\nabla f(\psi_t(w))$ initialized at any $\psi_0(w) = w \in U$ is assumed to remain and converge within U as $t \to \infty$.*

Under Assumption 1, we define the "projection" map $\Phi : U \to \mathcal{R}$ to River via the limit: $\Phi(w) := \lim_{t\to\infty} \psi_t(w)$. This ensures $\Phi(w) \in \mathcal{R}$ since the stationarity of the flow implies $P_s \nabla f = 0$.

Assumption 2 (Regularity, Smoothness and Conservation)**.** *Based on Assumption 1, we assume:*

1. *f is analytic in U. Let $(\cdot)|_s$ denote the restriction of an operator to the sharp subspace $\mathrm{Range}(P_s(w))$. For all $w \in U$, the eigenvalues of $\nabla^2 f(w)$, $F(w)$, $F(w)|_s$, $(F^{-\frac{1}{2}}(w)\nabla^2 f(w)F^{-\frac{1}{2}}(w))|_s$ lie in $[-L, L]$, $[\lambda_F, \rho]$, $[\lambda_{F_s}, \rho]$ and $[\lambda_{H_F,s}, L_{H_F,s}]$, respectively.*
2. *There exist constants $G, \varepsilon \geq 0$, such that for any $w \in U$, $\|\nabla f(w)\|_{F(w)^{-1}} \leq G$, $\|\nabla P_s(w)\|_{\mathrm{op}} \leq \varepsilon/G$ and $\|\nabla F(w)\|_{\mathrm{op}} \leq \varepsilon/G$. See (24) for detailed definition of the operator norm $\|\cdot\|_{\mathrm{op}}$.*
3. *There exists a time horizon $T_{max} > 0$ depending on $(\alpha, \beta_1, \{\eta_t\}_{t\geq 0}, \chi, \beta_2)$, such that for any initialization $w_0 \in U$ and $m_0 = 0$, the trajectory w_t of (4) remains strictly in U for all $t \in [0, T_{max}]$.*

Theorem 1. *Suppose Assumptions 1, 2 hold and $\dot{\eta}_t \leq 0$. Define $\iota_t = \min\{\alpha/2, \lambda_{H_{F,s}}\beta_1\eta_t\}$ and the forgetting kernel $K(t) = \exp\left(-\int_0^t \iota_s ds\right)$. There exists a constant $\varepsilon_0 > 0$ dependending on f and hyper-parameters in (4) such that for all $\varepsilon \leq \varepsilon_0$, the trajectory w_t governed by (4) satisfies the attraction bound $\|w_t - \Phi(w_t)\|_2^2 \lesssim K(t) + \varepsilon$, and the "projected" trajectory $z_t = \Phi(w_t)$ follows the flat-direction dynamics with $s_0 = 0$ and a perturbation $\epsilon_{z,t}$:*

$$\begin{cases} \dot{z}_t = -\eta_t \left(\chi P_{\mathcal{R}}(z_t)F(z_t)^{-1}(s_t + \beta_2 \nabla f(z_t)) + \epsilon_{z,t}\right), \\ \dot{s}_t = -\alpha s_t + \nabla f(z_t). \end{cases} \tag{5}$$

Finally, defining $b_t = \|P_{\mathcal{R}}(z_t)F(z_t)^{-1}\nabla f(z_t)\|^2_{F(z_t)} + \|P_{\mathcal{R}}(z_t)F(z_t)^{-1}s_t\|^2_{F(z_t)}$, the perturbation error satisfies $\|\epsilon_{z,t}\|^2_{F(z_t)} \lesssim K(t) + \varepsilon^2 b_t + \varepsilon \int_0^t K(\tau - t) b_\tau d\tau$.

The proof is in Lemmas 7 and 11. We treat ε as a small constant to enforce a slowly varying $\mathcal{R}$. Inspired by [3], we can take $\eta_t \equiv \eta$, $\alpha = 2\lambda_{F_{H,s}}^{1/2}\eta$, $\beta_1 = \lambda_{F_{H,s}}^{-1/2}$ to achieve a fast decay $K(t) = \exp(-\eta\lambda_{F_{H,s}}^{1/2} t)$ (*i*.e. fast convergence on sharp directions). In this case, Theorem 1 indicates that the dynamics (4) first converge to the $\mathcal{O}(\varepsilon^{1/2})$ neighborhood of $\mathcal{R}$ after a $\Theta(\eta^{-1}\lambda_{F_{H,s}}^{-1/2}\log(\varepsilon^{-1}))$ interval, then faithfully track the decoupled acceleration dynamics on $\mathcal{R}$ closely, where the tracking error exhibits a fading memory property, driven predominantly by the immediate history of the underlying dynamics. Next we clarify how χ and β_2 influence the convergence of the decoupled dynamics in $\mathcal{R}$, where the detailed proof is in Lemma 12.

Theorem 2. *Suppose Assumptions 1, 2 hold and $\dot{\eta}_t \leq 0$. Let $\Delta_f = f(z_0) - \inf f$. For the unperturbed dynamics in (5) with $\epsilon_{z,t} \equiv 0$, there exists a constant ε_1 (depending on f and hyper-parameters in (4)), such that for any $\varepsilon \leq \varepsilon_1$ and any $T \leq T_{\max}$, the trajectory z_t satisfies*

$$\frac{1}{T}\int_0^T \eta_t \|P_{\mathcal{R}}(z_t)F(z_t)^{-1}\nabla f(z_t)\|^2_{F(z_t)} dt \leq \frac{2\Delta_f}{\chi\beta_2 T}, \quad \text{and} \quad \frac{1}{T}\int_0^T \eta_t \|s_t\|^2_{F(z_t)} dt \leq \frac{2\Delta_f}{\chi\alpha T}.$$

Theorem 2 indicates that increasing both $\chi \geq 1$ and β_2 yields a tighter upper bound for the Riemannian gradient on $\mathcal{R}$ ($P_{\mathcal{R}}(z_t)F(z_t)^{-1}\nabla f(z_t)$) and the momentum s_t, thereby theoretically justifying the design motivation behind LITE.

8 Conclusion

We propose LITE, a generalized strategy to accelerate LLM pre-training by enhancing flat-direction training dynamics. LITE enhances state-of-the-art optimizers such as Muon and SOAP, achieving significantly lower terminal loss and exhibiting superior scaling behavior across various LLM pre-training tasks. Several limitations remain in this work. LITE introduces extra hyper-parameters that require a preliminary search on a proxy model, and the binary partition of sharp and flat directions leaves room for finer-grained precision. For future research, it is promising to explore dynamic hyper-parameter adaptation mechanisms in LITE, and to integrate it with other emerging optimizers.

9 Acknowledgment

This work is funded by the National Key Research and Development Program of China (No. 2024YFA1012902), and the National Natural Science Foundation of China (No. 12288101, W2441021). This research is also supported by AI for Science Institute, Beijing, China. Part of the computational resources were supported by the Center for Intelligent Computing and Song-Shan Lake HPC Cen-ter (SSL-HPC) in Great Bay University, Dongguan, China.

References

[1] N. Amsel, D. Persson, C. Musco, and R. M. Gower. The polar express: Optimal matrix sign methods and their application to the muon algorithm, 2025.

[2] V. I. Arnold. *Mathematical Methods of Classical Mechanics*. Springer-Verlag, New York, 2nd edition, 1989.

[3] H. Attouch, Z. Chbani, J. Fadili, and H. Riahi. First-order optimization algorithms via inertial systems with Hessian driven damping. *Mathematical Programming*, 193(1):113–155, 2022.

[4] J. Bernstein and L. Newhouse. Old optimizer, new norm: An anthology, 2024.

[5] J. Cohen, A. Damian, A. Talwalkar, J. Z. Kolter, and J. D. Lee. Understanding optimization in deep learning with central flows. In *The Thirteenth International Conference on Learning Representations*, 2025.

[6] V. Duruisseaux and M. Leok. A variational formulation of accelerated optimization on Riemannian manifolds. *SIAM Journal on Mathematics of Data Science*, 4(2):649–674, 2022.

[7] K. J. Falconer. Differentiation of the limit mapping in a dynamical system. *Journal of the London Mathematical Society*, s2-27(2):356–372, 1983.

[8] K. Frans, S. Levine, and P. Abbeel. A stable whitening optimizer for efficient neural network training, 2025.

[9] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[10] B. Ghorbani, S. Krishnan, and Y. Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2232–2241. PMLR, 09–15 Jun 2019.

[11] V. Gupta, T. Koren, and Y. Singer. Shampoo: Preconditioned stochastic tensor optimization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1842–1850. PMLR, 10–15 Jul 2018.

[12] A. Hägele, E. Bakouch, A. Kosson, L. Von Werra, M. Jaggi, et al. Scaling laws and compute-optimal training beyond fixed training durations. *Advances in Neural Information Processing Systems*, 37:76232–76264, 2024.

[13] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, and L. Sifre. Training compute-optimal large language models, 2022.

[14] S. Hu, Y. Tu, X. Han, C. He, G. Cui, X. Long, Z. Zheng, Y. Fang, Y. Huang, W. Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

[15] T. Huang, H. Hu, Z. Zhang, G. Jin, X. Li, L. Shen, T. Chen, L. Liu, Q. Wen, Z. Wang, and S. Liu. Stable-spam: How to train in 4-bit more stably than 16-bit adam, 2025.

[16] K. Jordan, Y. Jin, V. Boza, J. You, F. Cesista, L. Newhouse, and J. Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024.

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] T. T.-K. Lau, Q. Long, and W. Su. Polargrad: A class of matrix-gradient optimizers from a unifying preconditioning perspective, 2026.

[19] J. M. Lee. *Introduction to Riemannian Manifolds*, volume 176. Springer, New York, NY, 2 edition, 2018.

[20] K. Liang, L. Chen, B. Liu, and Q. Liu. Cautious optimizers: Improving training with one line of code, 2025.

[21] H. Liu, Z. Li, D. L. W. Hall, P. Liang, and T. Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *The Twelfth International Conference on Learning Representations*, 2024.

[22] J. Liu, J. Su, X. Yao, Z. Jiang, G. Lai, Y. Du, Y. Qin, W. Xu, E. Lu, J. Yan, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025.

[23] L. Liu, Z. Xu, Z. Zhang, H. Kang, Z. Li, C. Liang, W. Chen, and T. Zhao. Cosmos: A hybrid adaptive optimizer for memory-efficient training of llms, 2025.

[24] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[25] Y. Lu, L. Wang, and J. Liu. Understanding soap from the perspective of gradient whitening, 2025.

[26] J. Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

[27] J. Martens and R. Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2408–2417. PMLR, 07–09 Jul 2015.

[28] D. Morwani, I. Shapira, N. Vyas, eran malach, S. M. Kakade, and L. Janson. A new perspective on shampoo's preconditioner. In *The Thirteenth International Conference on Learning Representations*, 2025.

[29] M. Pagliardini, P. Ablin, and D. Grangier. The adEMAMix optimizer: Better, faster, older. In *The Thirteenth International Conference on Learning Representations*, 2025.

[30] T. Pethick, W. Xie, K. Antonakopoulos, Z. Zhu, A. Silveti-Falls, and V. Cevher. Training deep learning models with norm-constrained LMOs. In *Forty-second International Conference on Machine Learning*, 2025.

[31] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[32] B. Shi, S. S. Du, M. I. Jordan, and W. J. Su. Understanding the acceleration phenomenon via high-resolution differential equations. *Mathematical Programming*, 195(1-2), 2021.

[33] H.-J. M. Shi, T.-H. Lee, S. Iwasaki, J. Gallego-Posada, Z. Li, K. Rangadurai, D. Mudigere, and M. Rabbat. A distributed data-parallel pytorch implementation of the distributed shampoo optimizer for training neural networks at-scale, 2023.

[34] C. Si, D. Zhang, and W. Shen. Adamuon: Adaptive muon optimizer, 2025.

[35] M. Song, K. Ahn, and C. Yun. Does SGD really happen in tiny subspaces? In *The Thirteenth International Conference on Learning Representations*, 2025.

[36] M. Staib, S. Reddi, S. Kale, S. Kumar, and S. Sra. Escaping saddle points with adaptive gradient methods. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5956–5965. PMLR, 09–15 Jun 2019.

[37] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[38] W. Su. Isotropic curvature model for understanding deep learning optimization: Is gradient orthogonalization optimal?, 2025.

[39] W. Su, S. Boyd, and E. J. Candès. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. *Journal of Machine Learning Research*, 17(153):1–43, 2016.

[40] K. Team. Kimi k2: Open agentic intelligence, 2025.

[41] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models, 2023.

[42] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[43] N. Vyas, D. Morwani, R. Zhao, I. Shapira, D. Brandfonbrener, L. Janson, and S. M. Kakade. SOAP: Improving and stabilizing shampoo using adam for language modeling. In *The Thirteenth International Conference on Learning Representations*, 2025.

[44] J. Wang, M. Wang, Z. Zhou, J. Yan, W. E, and L. Wu. The sharpness disparity principle in transformers for accelerating language model pre-training. In *Forty-second International Conference on Machine Learning*, 2025.

[45] M. Wang, J. Wang, H. He, Z. Wang, G. Huang, F. Xiong, Z. li, W. E, and L. Wu. Improving generalization and convergence by enhancing implicit regularization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[46] M. Wang and L. Wu. The noise geometry of stochastic gradient descent. *Transactions on Machine Learning Research*, 2026.

[47] S. Wang, F. Zhang, J. Li, C. Du, C. Du, T. Pang, Z. Yang, M. Hong, and V. Y. F. Tan. Muon outperforms adam in tail-end associative memory learning, 2025.

[48] K. Wen, Z. Li, J. S. Wang, D. L. W. Hall, P. Liang, and T. Ma. Understanding warmup-stable-decay learning rates: A river valley loss landscape view. In *The Thirteenth International Conference on Learning Representations*, 2025.

[49] A. Wibisono, A. C. Wilson, and M. I. Jordan. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016.

[50] L. Wu, M. Wang, and W. J. Su. The alignment property of SGD noise and how it helps select flat minima: A stability analysis. In *Advances in Neural Information Processing Systems*, 2022.

[51] T. Xie, H. Luo, H. Tang, Y. Hu, J. K. Liu, Q. Ren, Y. Wang, W. X. Zhao, R. Yan, B. Su, C. Luo, and B. Guo. Controlled llm training on spectral sphere, 2026.

[52] X. Xie, P. Zhou, H. Li, Z. Lin, and S. Yan. Adan: Adaptive nesterov momentum algorithm for faster optimizing deep models, 2024.

[53] Z. Xie, W. Yin, and Z. Wen. Ode-based learning to optimize. *Mathematical Programming*, 2025.

[54] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[55] Z. Yang and J. Laaksonen. Principal whitened gradient for information geometry. *Neural Networks*, 21(2):232–240, 2008.

[56] Z. Yao, A. Gholami, K. Keutzer, and M. W. Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*, pages 581–590. IEEE, 2020.

[57] Z. Yao, A. Gholami, S. Shen, M. Mustafa, K. Keutzer, and M. W. Mahoney. Adahessian: An adaptive second order optimizer for machine learning, 2021.

[58] H. Yuan, Y. Liu, S. Wu, Z. Xun, and Q. Gu. MARS: Unleashing the power of variance reduction for training large models. In *Forty-second International Conference on Machine Learning*, 2025.

[59] Y. Zhang, C. Chen, T. Ding, Z. Li, R. Sun, and Z.-Q. Luo. Why transformers need adam: A hessian perspective. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[60] P. Zhou, J. Feng, C. Ma, C. Xiong, S. C. H. Hoi, and W. E. Towards theoretically understanding why sgd generalizes better than adam in deep learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 21285–21296, 2020.

[61] W. Zhou, B. Wang, W. Chen, and X. Cheng. Bsfa: Leveraging the subspace dichotomy to accelerate neural network training, 2025.

Appendix

Contents

A More Related Works **16**

A.1 Comparison with methods enhancing dynamics along flat directions. 16

A.2 Comparison with algorithmic frameworks for adaptive optimizers. 17

B Algorithm Details **17**

B.1 Details of Muon-LITE 17

B.2 Details of SOAP-LITE 18

B.3 Details of Hyperparameter Schedulers 19

C Experimental Details **19**

C.1 Experimental Details in Section 5.2 and Extended Experiments 20

C.2 Experimental Details in Sections 6 22

D Additional Experimental Results **24**

D.1 Verification of Isotropic Updates 24

D.2 Ablation Studies and Hyper-parameter Robustness 25

D.3 Supplementary Results on Dense Models 26

E Discussions on Momentum Formulations and their Continuous-Time ODE Counterparts **27**

E.1 Continuous-Time ODE Formulations 27

E.2 Heavy Ball Momentum, Nesterov Momentum, and Hessian Damping 27

E.3 Modeling AdEMAMix via a Third-Order ODE 29

F Foundations of Riemannian Geometry **30**

F.1 Riemannian Metric 30

F.2 The Levi-Civita Connection 31

F.3 Riemannian Gradient and Hessian 31

G Transforming the Second Order RISHD into a First Order System **32**

H Theoretical Analysis on Quadratic Objectives: An Illustrative Example **33**

I Some Common Adaptive Optimizers for LLM Pre-training **35**

I.1 Preconditioner-Estimated Riemannian Metrics for Common Adaptive Optimizers . 37

J Proof **38**

J.1 Landscape Analysis 38

J.2 Dynamics Analysis 44

J.3 Useful Lemmas 51

A More Related Works

A.1 Comparison with methods enhancing dynamics along flat directions.

Recent strategies accelerate LLM pre-training by assigning larger learning rates to flat components. [45] adjust learning rates based on an element-wise diagonal Fisher approximation, while [44] apply block-wise learning rates to blocks identified as flat via averaged sharpness. [61] further employs a matrix-level precision subspace detected via a lazy PCA-based estimator. Compared to these methods, LITE identifies flat subspaces with matrix-level precision using highly efficient numerical techniques that introduce no extra optimizer states. Crucially, prior studies focus exclusively on learning rate adjustment, a strategy LITE subsumes by simply setting $\chi \geq 1$. In contrast, we **also address the dynamics of momentum** in anisotropic landscapes. We show how momentum and preconditioners jointly influence training dynamics and **enhance momentum evolution** in flat directions by explicitly increasing Hessian damping.

Empirical comparison with existing flat-direction enhancements. We empirically compared LITE with the aforementioned flat-direction enhancement methods [45, 44, 61]. Because these prior methods were primarily evaluated using AdamW rather than matrix-based optimizers (like Muon and SOAP), we implemented them across both AdamW and Muon baselines for a comprehensive comparison.

Specifically, we applied the block-wise learning rate strategy proposed by [44] to our 250M LLaMA training setup (matching the experimental setting in Figure 4). Following their proposed sharpness principle and block-wise learning rate searching ranges, we searched for optimal hyperparameter ratios across the Embedding, QK, VO, and FFN blocks. As shown in Table 2, aggressively increasing the learning rates for the QK, VO, and FFN blocks actually degrades performance (increasing the loss). We hypothesize that this occurs because block-wise scaling is a coarse-grained operation: it indiscriminately amplifies both flat and sharp directions within each individual block, thereby exacerbating unstable oscillations along the sharp directions. Consequently, the best-performing configuration for this baseline achieves substantially weaker acceleration compared to Muon-LITE, which reaches a final validation loss of 2.078.

optimizer \ block wise lr	$(6, 4, 3, 2)$	$(4, 2, 2, 2)$	$(4, 1, 1, 1)$	$(1, 1, 1, 1)$ (baseline)
AdamW	6.540	2.295	**2.121**	2.129
Muon	3.673	2.212	**2.105**	2.109

Table 2: Results of Muon and AdamW using block wise learning rate on 250M LLaMA model trained on Pile. The optimal baseline configuration for AdamW, determined via hyperparameter search, is learning rate (lr)=0.003, $(\beta_1, \beta_2) = (0.9, 0.95)$, and weight decay=0.1. (a, b, c, d) means the learning rate for Embedding, QK, VO, FFN, RMSNORM blocks are $(a : b : c : d : 1)$, where 1 means the base learning rate. The best results are bold.

Regarding the methods proposed by [45] and [61], we found them difficult to scale to our 250M large-batch pre-training regime due to prohibitive memory overheads (notably, both prior works primarily verified their acceleration on models smaller than 230M parameters with smaller batch sizes). Specifically, [45] requires sampling from logits-generated distributions, which consumes over 31GB of additional Video Random Access Memory (VRAM) in our setup. Furthermore, its reliance on an Adam-style diagonal Fisher estimation to identify sharp directions is structurally incompatible with matrix-based optimizers like Muon. Similarly, [61] requires caching massive optimizer states (*e*.g., storing approximately 60 times the parameter count of the transformer matrices in their original LLaMA experiments) resulting in Out-of-Memory in our setting. In contrast, LITE introduces negligible memory overhead while seamlessly integrating with modern matrix-based optimizers.

A.2 Comparison with algorithmic frameworks for adaptive optimizers.

Recent works [30, 4, 51] interpret the update step of certain adaptive optimizers as the solution to a constrained optimization problem governed by operator norms:

$$U^* = \underset{\|U\|_{\text{op}} \leq 1}{\operatorname{argmin}} \langle U, M \rangle,$$

where M denotes the momentum at the current iterate, $\langle \cdot, \cdot \rangle$ is the Frobenius inner product, and $\|\cdot\|_{\text{op}}$ represents a specific operator norm. This formulation formally unifies methods like Muon and Lion, offering insights into their stability (e.g., by controlling the magnitude of activations during propagation). Since the geometry induced by this formulation is generally independent of the local curvature at the current iterate, its scope does not easily extend to methods like AdamW and SOAP, which utilize iterate-dependent preconditions and decouple momentum and preconditioner updates. It also leaves room for further explanations regarding the algorithmic design of accumulating momentum before preconditioning. Offering a different lens, our proposed Riemannian ODE framework unifies a broader class of adaptive optimizers. It uncovers the joint role of momentum and preconditioners in shaping training dynamics and extends naturally to complex momentum schemes (see Appendix E.3).

B Algorithm Details

B.1 Details of Muon-LITE

For Muon and Muon-LITE, we adopt the Newton-Schulz (NS) iteration termed Polar Express in [1], and set Newton–Schulz iterations to 6. This approach yields faster convergence to the theoretical optimum than the method in [16].

For the `embedding, norm, output` blocks, we use the same methods as in SOAP-LITE with $Q_l = I, Q_r = I$ to estimate the projection P_k to sharp directions. See more details in Section B.2. Next we introduce how to P_k for other blocks using Muon-LITE.

Estimating P_k via Efficient Composite Newton-Schulz. To efficiently estimate the sharp subspace projection P_k at iteration k, we employ a lightweight composite Newton-Schulz (NS) scheme. The process involves two main steps: constructing a filtering operator and dynamically adjusting the threshold. For any matrix A with Singular Value Decomposition (SVD) $A = U\Sigma V^\top$, we define $\text{NS}(A) = A(A^\top A)^{-\frac{1}{2}} = UV^\top$, where the inverse denotes the Moore-Penrose pseudo-inverse.

1. **Construction of the Filtering Operator.** We first define a dynamic threshold $\tau_k = l_k \|\widetilde{M}_k\|_F$, where l_k is a scaling factor. This threshold τ_k serves to delineate the sharp subspace: specifically, we select the eigenspace of $\widetilde{M}_k^\top \widetilde{M}_k$ associated with eigenvalues larger than τ_k^2 (singular values of $\widetilde{M}_k$ larger than τ_k). The adaptive update mechanism for l_k (and thus τ_k) will be detailed subsequently. We then compute the filtering operator $T_k \in \mathbb{R}^{m \times n}$ as:

$$T_k = \frac{1}{2}\,\text{NS}(\widetilde{M}_k) + \frac{1}{2}\,\text{NS}\left(\frac{\widetilde{M}_k}{\tau_k} - \text{NS}(\widetilde{M}_k)\right).$$

   The projection matrix onto the sharp subspace is subsequently obtained by $P_k = T_k^\top T_k$, the rationale for which is detailed in the subsequent parts ((6) and (7)). It is worth noting that this method for estimating P_k necessitates relatively high precision from the NS function. Consequently, we employ the Polar Express method [1] rather than the vanilla NS iterations used in [16].

2. **Dynamic Sharp Subspace Dimension Adjustment.** To ensure the rank of the projection P_k approximates the target dimension d_s, we update the scaling factor l_k based on the effective rank (measured by $\|P_k\|_F^2$):

$$l_{k+1} = \begin{cases} 1.05 \cdot l_k, & \text{if } \|P_k\|_F \geq \sqrt{d_s}, \\ 0.95 \cdot l_k, & \text{if } \|P_k\|_F < \sqrt{d_s}. \end{cases}$$

   This feedback loop effectively maintains τ_k near the d_s-th largest singular value of $\widetilde{M}_k$.

Theoretical Intuition. To understand why T_k acts as a subspace filter, consider the SVD of $\widetilde{M}_k = U\Sigma V^\top$. Let the singular values be ordered as $\sigma_1 \geq \cdots \geq \sigma_j > \tau_k > \sigma_{j+1} \geq \cdots \geq 0$. Since the Newton-Schulz iteration converges to the matrix sign function, we have $\mathrm{NS}(\widetilde{M}_k) = UV^\top$. Consequently, the input to the second NS term becomes:

$$\frac{\widetilde{M}_k}{\tau_k} - \mathrm{NS}(\widetilde{M}_k) = U \,\mathrm{diag}\left(\frac{\sigma_1}{\tau_k} - 1, \ldots, \frac{\sigma_r}{\tau_k} - 1\right) V^\top. \tag{6}$$

Applying $\mathrm{NS}(\cdot)$ to (6) maps positive entries (where $\sigma_i > \tau_k$) to $+1$ and negative entries (where $\sigma_i < \tau_k$) to -1. Therefore, we have

$$T_k = \frac{1}{2}\mathrm{NS}(\widetilde{M}_k) + \frac{1}{2}\mathrm{NS}\left(\frac{\widetilde{M}_k}{\tau_k} - \mathrm{NS}(\widetilde{M}_k)\right) = U\,\mathrm{diag}(1, \ldots, 1, 0, \ldots, 0)V^\top. \tag{7}$$

Computational Efficiency of Muon-LITE. Overall, Muon-LITE requires only two additional NS functions per step compared to vanilla Muon, which can be executed in parallel. Since NS iterations constitute a nearly negligible fraction of the computational cost in large-batch pre-training (dominated by gradient computation), this slight overhead is virtually imperceptible in practice. Table 3 reports the average per-iteration wall-clock times of Muon and Muon-LITE across model sizes from 0.25B to 1.3B, with torch.compile enabled for the model and Newton–Schulz iterations sharded across GPUs. The results indicate that Muon-LITE introduces only a modest time overhead. Since the extra overhead of LITE is independent of batch size, it diminishes with larger batches, as gradient computation will occupy significantly more time, making this benefit particularly relevant for large-scale industrial training with token batch sizes of tens of millions or more. For example, training a LLaMA-1.3B model on the Pile dataset (global batch size 8192, micro batch size 16, sequence length 1024, on $8\times$ A800-80GB GPUs) yields a throughput of 100.4k tokens/s for Muon-LITE versus 101.5k tokens/s for Muon. This reflects a throughput drop of only $\approx 1\%$. Furthermore, efficiency can be enhanced via system-level optimizations, such as kernel fusion for the NS iterations.

Table 3: Average per-iteration wall-clock times (seconds) of Muon and Muon-LITE across model sizes.

Model	**0.25B**	**0.5B**	**1.3B**
Muon	0.957	1.382	4.908
Muon-LITE	0.989 (+3.3%)	1.410 (+2.0%)	4.980 (+1.4%)

B.2 Details of SOAP-LITE

In SOAP-LITE, we employ a dynamic thresholding strategy generating a smoothed projection onto the sharp directions to enhance training stability. To avoid the computational latency and memory overhead associated with exact `top_k` operations in PyTorch, especially for large matrices, we approximate the top-k thresholds via adaptive scalar variables, similar to the approach in Muon-LITE. We define d_s as the target dimension of the sharp subspace, and d_{smooth} as the dimension of the transitional subspace bridging the sharp and flat directions. We maintain two dynamic scalars, l_k^s (controlling the sharp boundary) and l_k^{smooth} (controlling the smooth boundary), initialized as $l_0^s = 1.0$ and $l_0^{\text{smooth}} = 0.5$.

At the k-th iteration, we estimate the thresholds for the top-d_s and top-$(d_s + d_{\text{smooth}})$ elements based on the mean magnitude of the tensor V_k: $\tau_k^s = l_k^s \,\mathrm{mean}(V_k)$ and $\tau_k^{\text{smooth}} = l_k^{\text{smooth}} \,\mathrm{mean}(V_k)$. The smoothed projection matrix $P_k \in \mathbb{R}^{m\times n}$ is then constructed element-wise as:

$$(P_k)_{i,j} = \begin{cases} 1, & \text{if } (V_k)_{i,j} \geq \tau_k^s, \\ \dfrac{(V_k)_{i,j} - \tau_k^{\text{smooth}}}{\tau_k^s - \tau_k^{\text{smooth}}}, & \text{if } \tau_k^{\text{smooth}} \leq (V_k)_{i,j} < \tau_k^s, \\ 0, & \text{if } (V_k)_{i,j} \leq \tau_k^{\text{smooth}}. \end{cases}$$

Then we dynamically adjust the coefficients l_k^s and l_k^{smooth} based on the current sparsity levels:

$$l_{k+1}^s = \begin{cases} 1.05 \cdot l_k^s, & |\{V_{i,j} : V_{ij} > \tau_k^s\}| \geq d_s, \\ 0.95 \cdot l_k^s, & |\{V_{i,j} : V_{i,j} > \tau_k^s\}| < d_s, \end{cases}$$

and

$$l_{k+1}^{\text{smooth}} = \begin{cases} 1.05 \cdot l_k^{\text{smooth}}, & |\{V_{i,j} : V_{ij} > \tau_k^{\text{smooth}}\}| \geq d_s + d_{\text{smooth}}, \\ 0.95 \cdot l_k^{\text{smooth}}, & |\{V_{i,j} : V_{i,j} > \tau_k^{\text{smooth}}\}| < d_s + d_{\text{smooth}}, \end{cases}$$

where $|\cdot|$ denotes the cardinality of the set. Finally, we enforce the constraint $l_{k+1}^{\text{smooth}} \leftarrow \min\{0.95 l_{k+1}^s, l_{k+1}^{\text{smooth}}\}$ to ensure that the smoothing threshold remains strictly lower than the sharp threshold ($l_k^{\text{smooth}} < l_k^s$). Our approach eliminates the use of inefficient top-k functions, thereby incurring almost no additional computational or temporal burden.

B.3 Details of Hyperparameter Schedulers

We keep β_1 and β_2 fixed throughout training and impose a schedule on χ in LITE. When the sharp directions use the `cos` learning rate scheduler, we also use the `cos` scheduler for the flat-direction learning rate. Specifically, we define the `cos` scheduler as

$$\cos(\eta_{\min}, \eta_{\max}, T_{\max}, T_{\text{warm}}, t) = \begin{cases} \dfrac{t\eta_{\max}}{T_{\text{warm}}}, & 0 \leq t < T_{\text{warm}}, \\ \eta_{\min} + \dfrac{1}{2}(\eta_{\max} - \eta_{\min}) \left(1 + \cos\left(\dfrac{\pi(t - T_{\text{warm}})}{T_{\max} - T_{\text{warm}}}\right)\right), & T_{\text{warm}} \leq t \leq T_{\max}. \end{cases}$$

If the flat directions adopt the learning-rate scheduler $\cos(\eta_{\min}, \eta_{\max}, T_{\max}, T_{\text{warm}}, \cdot)$, then the flat-direction learning-rate ratio χ (i.e., $\chi(t)$) follows the scheduler

$$\chi(t) = \min \left\{ \underbrace{1 + \frac{\max\{t - T_{\text{warm}}, 0\}}{T_{\max} - T_{\text{warm}}} \frac{\tilde{\chi} - 1}{T_{\text{warm}}^{\text{flat}}}}_{\text{warmup}}, \underbrace{\frac{\cos(\eta_{\min}, \eta_{\max}\tilde{\chi}, T_{\max}, T_{\text{warm}}, t)}{\cos(\eta_{\min}, \eta_{\max}, T_{\max}, T_{\text{warm}}, t)}}_{\text{decay}} \right\},$$

where $\tilde{\chi} \geq 1$ is the amplification ratio (e.g., 2.0 in the experiments). We set $T_{\text{warm}}^{\text{flat}} \geq T_{\text{warm}}$ so that χ has a longer warmup period and training remains stable during the initial phase. We set $T_{\text{warm}}^{\text{flat}} = 0.5 T_{\max}$ in all experiments using the `cos` learning rate scheduler.

C Experimental Details

Models. We utilize two popular classes of LLM models for our pre-training experiments:

- **LLaMA.** LLaMA [41] is a widely adopted dense decoder-only Transformer architecture. It employs Rotary Positional Embeddings (RoPE) [37], Swish-Gated Linear Units (SwiGLU), and Root Mean Square Layer Normalization (RMSNorm). In this work, we pre-train LLaMA models with sizes ranging from 130M to 1.3B parameters. Detailed model configurations are provided in Table 4.
- **QwenMoE.** Qwen2MoE [54] is a prominent open-source Mixture-of-Experts (MoE) decoder-only Transformer. In contrast to LLaMA, Qwen2MoE integrates a hybrid attention mechanism (combining sliding window and full attention) alongside its MoE architecture. For our experiments, we disable sliding window attention given the relatively short context length. We configure the model to activate 4 experts per token. Auxiliary losses including z-loss with coefficient 0.001 and load balancing loss with coefficient 0.01 are used to make sure stable training. Refer to Table 5 for comprehensive configuration details.

Datasets. We conduct pre-training on the following datasets:

- **C4** [31]. The Colossal Clean Crawled Corpus (C4) dataset is employed for our small-to-medium scale pre-training experiments. We use the T5 tokenizer with a vocabulary size of 32,100.
- **The Pile** [9]. We utilize The Pile for large-scale (or larger-context) pre-training tasks. For this dataset, we adopt the LLaMA-2 tokenizer [42] with a vocabulary size of 32,000.

Table 4: Dense model configurations and optimally-tuned peak learning rates for Muon and SOAP.

Acronym	Size	d_{model}	d_{FF}	n_head	depth	`lr_max` of Muon	`lr_max` of SOAP
LLaMA (0.13B)	134M	768	2048	12	12	5e-3 (on C4)	-
LLaMA (0.25B)	247M	768	2560	16	24	3e-3 (on Pile)	3e-3 (on Pile)
LLaMA (0.5B)	518M	1280	3456	20	22	2e-3 (on Pile)	2e-3 (on Pile)
LLaMA (1.3B)	1339M	2048	5461	32	24	1e-3 (on Pile)	-

Table 5: MoE model configurations and optimally-tuned peak learning rates for Muon on Pile.

Acronym	Size	Activated Size	d_{model}	d_{FF}	n_head	depth	n_experts	`lr_max`
QwenMoE (1B)	1040M	297M	768	3072	12	15	32	2e-3

Sequence Packing/Batching. For the Pile dataset, we employ a standard *sequence packing* strategy: documents are tokenized, concatenated into a continuous stream, and then segmented into fixed-length sequences of size L (depending on the dataset). This approach minimizes padding overhead and ensures a near-constant number of effective tokens per batch. Training progress is measured in terms of total tokens, with evaluation performed on the official validation splits.

LR schedulers. We evaluate two popular learning-rate (LR) scheduling strategies with 1,000 warm-up steps in all pre-training experiments:

- `cos` (cosine scheduler) [42]: a linear warm-up to peak `lr_max`, followed by cosine decay to a terminal LR `lr_min`. Following [13], we set `lr_min` $= 0.1 \times$ `lr_max`.
- `wsd` (warmup-stable-decay scheduler) [14, 12]: a linear warm-up to `lr_max`, followed by a stable phase keeping LR at `lr_max` (up to 80% of total steps), and then a linear decay to 0 over the final 20% steps.

C.1 Experimental Details in Section 5.2 and Extended Experiments

In this subsection, we demonstrate that gradients (first-order information) can effectively approximate the top eigenspaces of the Hessian (second-order information) for matrix block parameters in language models.

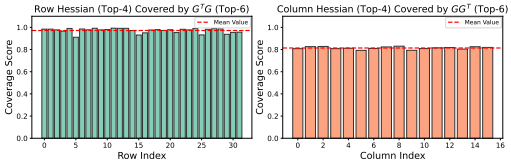
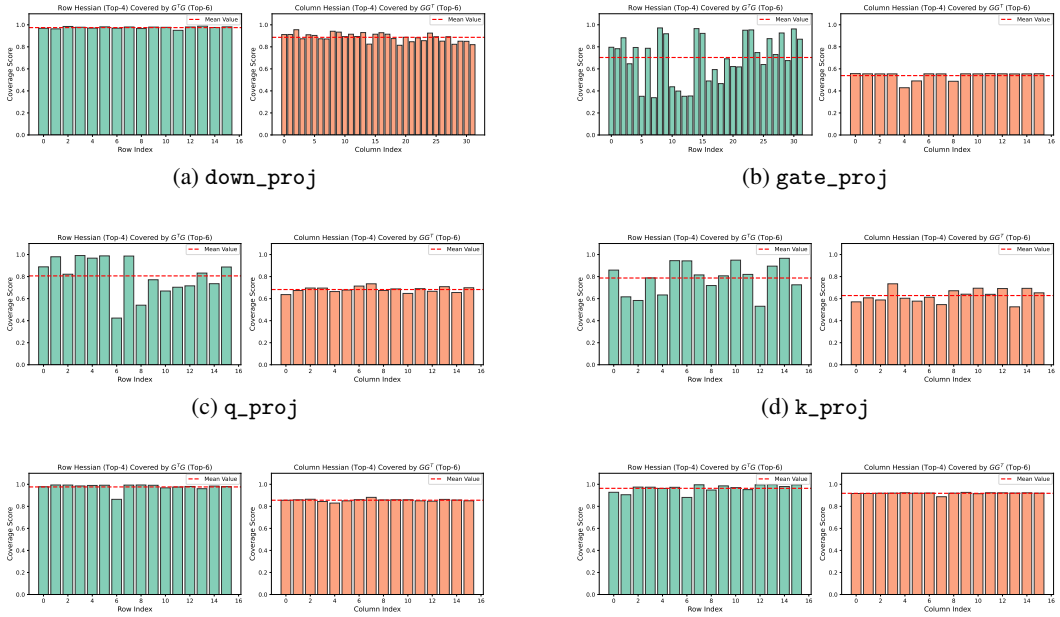


Figure 6: Coverage of the top eigenspaces of row (column) Hessians by those of $G^\top G$ ($GG^\top$) for the `up_proj` block of an FFN layer. A higher score indicates a greater degree of containment.

Validation on a Toy Model Hessian-noise alignment phenomena have been extensively investigated and verified in simplified models, both theoretically and empirically [46, 50]. To validate this in our context, we conduct an experiment on a small LLAMA2 model with $d_{\text{model}} = 16, d_{\text{ffn}} = 32,$ n_head $= 4,$ depth $= 4,$ vocabulary size $= 8,$ #params $= 10,640$. The model is trained using AdamW for 100 steps with a token batch size $8 \times 8 = 64$ and lr $= 0.001$. For Q,K,V,O, FFN (up, gate, down) blocks, we compute the Hessian eigenvalue distribution within each block (Figures 2 and 8), test the alignment degree of each row's Hessian with $G^\top G$, and each column's Hessian with $GG^\top$ (Figures 6 and 7). The results shows that for a small dimension $d_s = 4$ ($4/32$ and $4/16$), the top-d_s eigenspaces of the Hessian (for rows/columns) are well covered by a subspace of $G^\top G$ ($GG^\top$) of dimension only slightly larger than d_s (Figures 6 and 7), where G is the stochastic gradient matrix.

Validation on a 120M LLaMA Model Now we verify the block-wise Hessian-gradient alignment in a 120M dense LLaMA model with $d_{\text{model}} = 768$, $d_{\text{FFN}} = 2048$, $n_{\text{head}} = 12$, and $n_{\text{layer}} = 6$, trained with a batch size of 480 and a sequence length of 1024. We fix a matrix parameter block $W \in \mathbb{R}^{m \times n}$, and use $\text{vec}_{\text{row}}(W)$ to denote row-major vectorization: $w = \text{vec}_{\text{row}}(W) \in \mathbb{R}^{mn}$. The corresponding gradient and Hessian on a mini-batch $\mathcal{B}$ is

$$G \in \mathbb{R}^{m \times n}, \quad H_W = \nabla_w^2 L_{\mathcal{B}} \in \mathbb{R}^{mn \times mn}.$$

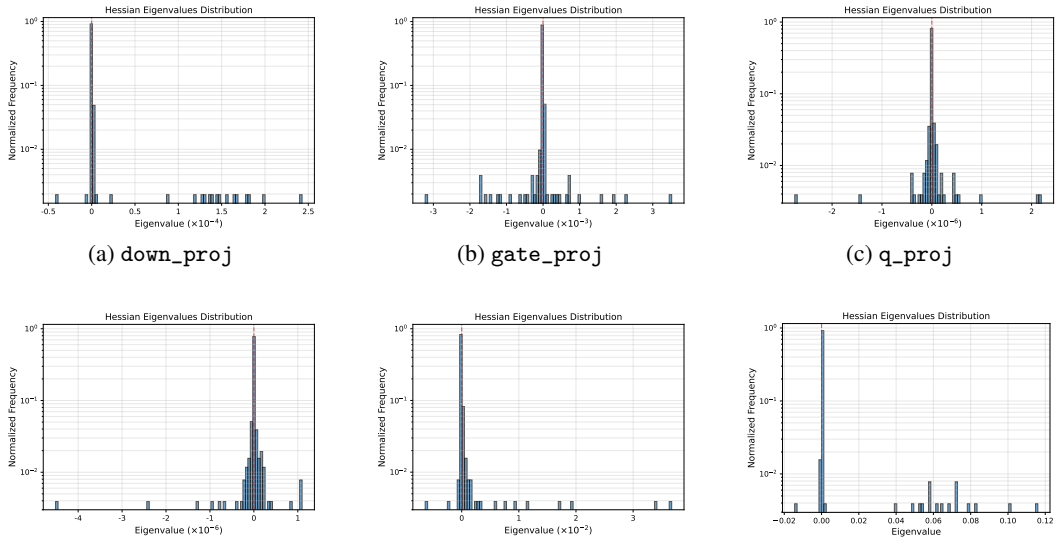


(a) `down_proj` (b) `gate_proj`

(c) `q_proj` (d) `k_proj`

(e) `v_proj` (f) `o_proj`

Figure 7: Coverage scores across row and column indices of different blocks in the toy LLaMA2 model. The results of the `up_proj` are in Figure 6.



(a) `down_proj` (b) `gate_proj` (c) `q_proj`

(d) `k_proj` (e) `v_proj` (f) `o_proj`

Figure 8: Hessian eigenvalue distribution of different blocks in the toy LLaMA2 model. The results of the `up_proj` are in Figure 2.

1. **Matrix-free block-Hessian eigenspace estimation.** We avoid explicitly constructing H_W. For a matrix direction $D \in \mathbb{R}^{m\times n}$, by $\mathrm{vec}_{\mathrm{row}}\left(\mathcal{H}_W[D]\right) = H_W \mathrm{vec}_{\mathrm{row}}(D)$, we have $\mathcal{H}_W[D] = \nabla_W \langle G, D\rangle_F$. We apply restarted Lanczos iterations to this block-restricted operator. Starting from a normalized random vector q_1, each iteration computes

$$
\begin{aligned}
z_j &= H_W q_j, \\
\alpha_j &= q_j^\top z_j, \\
r_j &= z_j - \alpha_j q_j - \beta_{j-1} q_{j-1}, \\
\beta_j &= \|r_j\|_2,
\end{aligned}
$$

$$q_{j+1} = r_j/\beta_j.$$

After p iterations, let $Q_p = [q_1, \ldots, q_p]$ denote the Krylov basis. The projected Hessian is $T_p = Q_p^\top H_W Q_p$. We compute its eigendecomposition $T_p y_i = \widehat{\lambda}_i y_i$ and map the resulting Ritz vectors back to the original parameter space as $\widehat{e}_i = Q_p y_i$.

We retain the k algebraically largest Ritz values and define the **estimated top-k Hessian eigenspace as**

$$Q_H = [\widehat{e}_1, \ldots, \widehat{e}_k] \in \mathbb{R}^{mn \times k}.$$

Each Ritz vector can equivalently be represented as a matrix $E_i = \mathrm{vec}_{\mathrm{row}}^{-1}(\widehat{e}_i) \in \mathbb{R}^{m \times n}$. Convergence is checked using the relative Ritz residual

$$\rho_i = \frac{\|H_W \widehat{e}_i - \widehat{\lambda}_i \widehat{e}_i\|_2}{1 + |\widehat{\lambda}_i|}.$$

2. **Top subspace of the gradient.** Using the same fixed mini-batch, we compute the singular value decomposition of the gradient $G = U\Sigma V^\top$. Let $U_s = U_{[:,1:s]}$, $V_s = V_{[:,1:s]}$ denote the leading s left and right singular vectors. Then the top left/right subspaces for the gradient are

$$B_{\text{left}} = U_s \otimes I_n \in \mathbb{R}^{mn \times sn}, \quad B_{\text{right}} = I_m \otimes V_s \in \mathbb{R}^{mn \times ms}.$$

For the up, gate, down, q, k, v, and o matrices of a given layer, Figure 3 reports the coverage scores between the top gradient subspaces and the Hessian subspaces. The results show that **the top right gradient subspaces achieve well coverage of the top Hessian subspaces**, while the top left gradient subspaces also exhibit a certain degree of alignment with the top Hessian subspaces.

Measure Coverage Degree between Two Subspaces. Let A and B be two linear subspaces of $\mathbb{R}^d$ with $\dim(A) \leq \dim(B)$. Let $P_A = [u_1, ..., u_{k_A}] \in \mathbb{R}^{d \times k_A}$ and $P_B = [v_1, ..., v_{k_B}] \in \mathbb{R}^{d \times k_B}$ be the matrices consisting of the orthonormal bases of A and B, respectively. We define the Coverage Score to quantify the extent to which A is contained within B:

$$\text{Coverage Score}(A, B) = \frac{1}{k_A} \|P_A^\top P_B\|_* = \frac{1}{k_A} \sum_{i=1}^{k_A} \cos\theta_i, \tag{8}$$

where $\|\cdot\|_*$ denotes the nuclear norm, and θ_i represent the i-th principal angle between subspaces A and B. By definition, the Coverage Score lies in the interval $[0, 1]$. A score of 1 indicates that $A \subseteq B$, while a score of 0 implies that A is orthogonal to B ($A \perp B$). A value closer to 1 suggests a higher degree of containment of A within B.

C.2 Experimental Details in Sections 6

Muon Baselines. We adopt the standard Muon configuration described in [22]. Specifically, we apply Muon specifically to 2D matrix blocks within transformer layers, while using AdamW for all other parameters (RMSNorm (`norm`), Embedding (`emb`) and the Output (`out`)). We set the learning rate scaling factor to $c = 0.2\sqrt{\max\{m, n\}}$ for blocks that use Muon with shape $\mathbb{R}^{m \times n}$, ensuring the update RMS norm aligns with that of AdamW. The baseline is configured with Nesterov momentum using a decay coefficient $\theta_{\text{muon}} = 0.95$, weight decay $\lambda = 0.1$, and a gradient clipping threshold of 1.0.

SOAP Baselines. We maintain the same weight decay and gradient clipping settings as in Muon. Following [43], we set the hyper-parameters to $\theta = 0.95$, $\beta_v = 0.99$, and $k = 10$. regarding the shampoo momentum, although [43] recommends $\beta_{\text{shampoo}} = 0.99$, we compared $\beta_{\text{shampoo}} \in \{0.99, 0.95\}$ and selected 0.95, as it yielded lower validation losses in our experiments.

Learning Rate Schedules. For the `cos` schedule, following [13], the minimum learning rate `lr_min` is set to $0.1 \times$ `lr_max`. For the `wsd` schedule, we allocate the warmup and stable phases to the first 80% of total iterations, followed by a linear decay to 0 over the remaining 20%. We set the warm-up steps to 1000 for all pre-training experiments.

Additionally,

- **C4 pre-training.** We use a sequence length of 256 and a global batch size of 3,600. The total number of training tokens is set to approximately 80 times the number of model parameters, corresponding to $4\times$ the Chinchilla scaling law [13]. The training process includes 1,000 warm-up steps. We perform a grid search for the maximum learning rate `lr_max` over the set $\{\texttt{1e-3}, \texttt{2e-3}, \texttt{3e-3}, \texttt{5e-3}, \texttt{7e-3}\}$. The resulting optimal learning rates for each model are detailed in Table 4. The relevant experiments are performed on four NVIDIA A100 GPUs, each with 80 GB of memory.
- **Pile pre-training.** We set the sequence length to 1,024. The batch size is set to 1,024 for the 0.25B and 0.5B models, and 2,048 for the 1.3B model. The total number of training tokens is approximately 40 times the number of model parameters (corresponding to $2\times$ the Chinchilla scaling law [13]), which includes 1,000 warm-up steps. A grid search for `lr_max` is performed over the set $\{\texttt{7e-4}, \texttt{1e-3}, \texttt{2e-3}, \texttt{3e-3}, \texttt{5e-3}, \texttt{7e-3}\}$. The optimal learning rates for each model are detailed in Tables 4 and 5. The relevant experiments are performed on four (model <1 B) or eight (model ≥ 1 B) NVIDIA H100 GPUs, each with 80 GB of memory.

Hyper-parameter tuning for Muon-LITE. For LLaMA models, we identified the optimal hyper-parameters $\{\chi, \beta_1, \beta_2\}$ via searching on the 250M model (Pile) and applied them uniformly to the 130M (C4), 500M (Pile), and 1.3B (Pile) experiments. Specifically, fixing the base learning rate to Muon's optimal value, we searched the sharp subspace dimension ration $r_s \in \{0.1, 0.2\}$, and set $d_s = r_s \min\{m, n\}$ for `q,k,v,o,ffn` blocks of size $\mathbb{R}^{m\times n}$, and $d_s = r_s mn$ for `emb, norm` blocks, with all of them using the same ratio r_s. For AdamW blocks, we set $\beta_{1,2} = 0.0$ for brevity, and $d_{\text{smooth}} = 0.1mn$ (the final loss is insensitive to this parameter). Following [44], we exclude the `output` block from LITE (*i*.e. treating the entire block as sharp directions), as accelerating it offers minimal gains and may risk instability. To avoid expensive searching for the hyper-parameters in LITE, **we use an efficient sequential and greedy search on a 250M LLaMA model** as follows. This practice of **searching for hyperparameter configurations on small models and directly scaling them up to larger models and longer training procedures is also common in recent optimizer studies**, such as [44], [58], [45]. We denote the value of χ for blocks using AdamW and Muon by $\chi_{\text{adamw}}, \chi_{\text{muon}}$ respectively.

Step 1: Fix d_s to the top-20% of the total dimensions and set $\beta_1 = 0$. Search over $\chi \in \{2, 4\}$ for blocks using Muon and AdamW, respectively.

Step 2: Based on the best configuration from Step 1, search over $\beta_2 \in \{0.5, 1.0, 2.0\}$.

Step 3: Based on the best configuration from Step 2, test $\beta_1 = -0.25$ to determine whether it yields further improvement.

Step 4: Finally, reduce d_s to the top-10% of the total dimensions to examine potential additional gains.

Here we permit $\beta_1 \leq 0$ as Muon is inherently equipped with Nesterov momentum. As a result, the parameter β_1 here is not required to be positive and functions primarily as an extra corrective term. The entire search procedure requires a maximum of 9 individual searches. Finally, we selected $r_s = 0.1$ and the specific settings in LITE variants for LLaMA models are:

- Muon-LITE-H: $(\beta_1, \beta_2) = (-0.25, 2.0)$.
- Muon-LITE-L: $\chi_{\text{adamw}} = 4$, $\chi_{\text{muon}} = 2$.
- Muon-LITE: Adopts th χ settings from Muon-LITE-L and set $(\beta_1, \beta_2) = (-0.25, 1.0)$.

For Muon-LITE in the QwenMoE experiment, we set $r_s = 0.1$ and $\beta_1 = 0.0$ for brevity. We searched $\chi_{\text{adamw}} \in \{4, 8\}$ (while maintaining $\chi_{\text{muon}} = 1$ for others) and $\beta_2 \in \{1.0, 2.0\}$ for blocks using Muon. The final configuration adopted was $\chi_{\text{adamw}} = 8$ and $\beta_2 = 1.0$.

Remark 3 (The value of β_1)**.** *Taking $\beta_1 < 0$ in Algorithm 1 would violate the convergence condition. In practice, we allow $\beta_1 < 0$, as long as the "effective damping coefficient β_1" remains positive. The apparent discrepancy arises from a notational difference: the β_1 notation in Algorithm 1 (denoted as β_1 itself) differs from that in the implemented Muon-LITE update (denoted as $\beta_1^{\textit{Muon-LITE}}$), where the latter already includes an intrinsic Nesterov (Hessian-damping) term, making β_1 in Muon-LITE merely an additive correction. The Muon-LITE update is*

$$U_k = \widetilde{M}_k((\widetilde{M}_k)^\top M_k)^{-\frac{1}{2}}(P_k + \chi(I_n - P_k)) + G_k(G_k^\top G_k)^{-\frac{1}{2}}(\beta_1^{\textit{Muon-LITE}} P_k + \chi\beta_2(I_n - P_k)).$$

Expanding the sharp-direction part, the operator acting on the sharp-direction gradient G_k is

$$A_k := \theta^{-1}((\widetilde{M}_k)^\top \widetilde{M}_k)^{-\frac{1}{2}} + \beta_1^{\textit{Muon-LITE}}(G_k^\top G_k)^{-\frac{1}{2}}.$$

Approximately we have $(\widetilde{M}_k)^\top \widetilde{M}_k \approx (1-\theta^2)^{-1} G_k^\top G_k$ (Section I.1). Thus for $\theta = 0.95$ and $\beta_1^{\textit{Muon-LITE}} = -0.25$, the effective "$\beta_1$" in Algorithm 1 is positive: $\theta^{-1}(1-\theta^2)^{1/2} + \beta_1^{\textit{Muon-LITE}} > 0.3 - 0.25 > 0$. We explored negative $\beta_1^{\textit{Muon-LITE}}$ values only to add flexibility in fine-tuning the sharp-direction adaptive damping. Compared with $\beta_1^{\textit{Muon-LITE}} = 0$, the value -0.25 yields a modest and gradually diminishing loss reduction during the early-to-late training stages (from about 0.003 at 2k steps down to 0.001 at 10k steps). Thus one can also set $\beta_1^{\textit{Muon-LITE}} = 0$ for simplicity and still achieve comparable performance. We also note that setting $\beta_1^{\textit{Muon-LITE}} = -0.5$ drives the effective coefficient below zero and degrades the final loss by approximately 0.01, which further confirms that our practical choice stays within the theoretically valid region.

Extra hyper-parameter tuning for the Muon baseline. The hyper-parameters we used for the Muon baseline ($\lambda = 0.1$, $\theta_{\text{muon}} = 0.95$) are standard in the related literature. As discussed in the previous section, we conducted hyperparameter search for Muon-LITE. This search was performed on relatively short runs with a small model, and the optimized hyper-parameters were directly transferred to larger models and longer training runs. Nevertheless, for a fairer comparison, we additionally tuned a more extensive set of hyper-parameters for the Muon baseline, including block-wise learning rate, weight decay coefficient, and momentum decay coefficient, to explore its performance ceiling. After tuning the block-wise learning rate (Table 2), weight decay, and momentum decay coefficients (Table 6), we observed that the best configuration only outperforms the baseline by 0.005, which is substantially smaller than the gain delivered by Muon-LITE (0.031). This indicates that the standard Muon baseline hyper-parameters are already near-optimal, and the improvements from Muon-LITE stem from its algorithmic design.

θ_{muon} \ λ	0.05	0.1	0.2
0.9	2.104 (best, -0.005)	2.105	2.105
0.95	2.110	2.109 (base)	2.109
0.98	2.155	2.155	2.154

Table 6: Results of tuning momentum decay coefficient θ_{muon} and weight decay coefficient λ on Muon (250M LLaMA model trained on Pile).

Hyper-parameter tuning for SOAP-LITE. Similar to Muon-LITE. we identified the optimal hyper-parameters $\{\chi, \beta_1, \beta_2\}$ via grid search on the 250M model (Pile) and applied them to other scenarios. Specifically, fixing the base learning rate to SOAP's optimal value, we searched the sharp subspace dimension ration $r_s \in \{0.1, 0.2\}$, and set $d_s = r_s mn$ for all blocks, with all of them using the same ratio r_s. The search space for other coefficients is $\beta_1 = 0.0$, $\beta_2 \in \{0.25, 0.5, 1.0\}$, and $\chi \in \{2.0, 4.0\}$, where we use similar greedy and sequential search as for Muon-LITE to avoid exhausted grid search. We set $d_{\text{smooth}} = 0.1mn$ uniformly (the final loss is insensitive to this parameter). Following [44], we exclude the `output` block from LITE (*i*.e. treating the entire block as sharp directions), as accelerating it offers minimal gains and may risk instability. Finally, we selected $r_s = 0.2$ and the specific settings in SOAP-LITE variants for LLaMA models are:

- SOAP-LITE-H: $(\beta_1, \beta_2) = (0.0, 0.5)$.
- SOAP-LITE-L: $\chi_{\text{adamw}} = 4$, $\chi_{\text{soap}} = 2$.
- SOAP-LITE: Adopt the χ settings from SOAP-LITE-L and set $(\beta_1, \beta_2) = (0.0, 0.25)$.

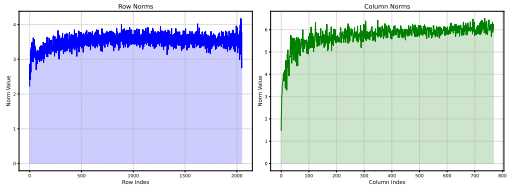
D Additional Experimental Results

D.1 Verification of Isotropic Updates

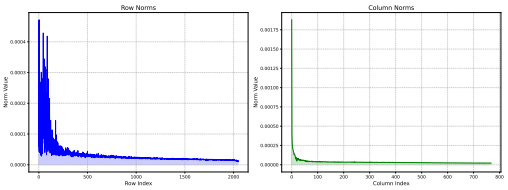
Isotropy in optimization landscapes manifests not only as strict per-step normalization but also as distributional isotropy (e.g., conforming to a standard Gaussian). Sign-based algorithms like signSGD

and Muon explicitly enforce strict isotropy by clipping the update magnitude along each eigen-direction to a fixed constant per step. Methods like AdamW and SOAP decouple the preconditioner and momentum update, achieving isotropy through their statistical distributions.

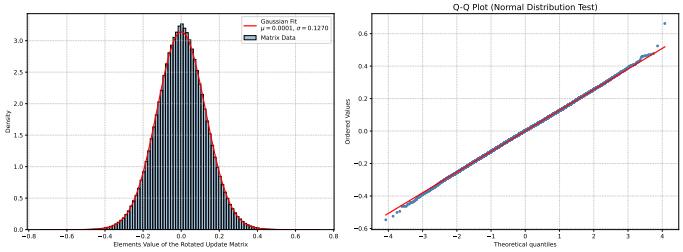
We empirically demonstrate this distributional isotropy using SOAP as an example. Let (m, g) and M, G represent the vector-form and matrix-form momentum and stochastic gradient. SOAP computes a whitened update $u = F^{-1}m$ (vector-form) with $U = Q_l \frac{Q_l^\top M Q_r}{\sqrt{V_{\text{rot}}}} Q_r^\top$ (matrix-form), where Q_l, Q_r are orthogonal matrices. Here, $F = (Q_r \otimes Q_l)(\operatorname{diag} v_{\text{rot}})^{\frac{1}{2}}(Q_r^\top \otimes Q_l^\top)$ effectively estimates the uncentered covariance $\mathbb{E}[gg^\top]$ (see discussions in Appendix I.1). This whitening process drives u toward distributional isotropy, meaning its covariance tends to aI for some constant a. We verify this using a FFN block from a 130M LLaMA model. In the SOAP-rotated eigenspace, the update $U_{\text{rot}} = \frac{Q_l^\top M Q_r}{\sqrt{V_{\text{rot}}}}$ (matrix-form) demonstrates nearly uniform row and column norms (Figure 9a), with its individual elements closely follow an independent Gaussian distribution (Figure 9c). Notably, the unwhitened raw momentum $M_{\text{rot}} = Q_l^\top M Q_r$ fails to exhibit such isotropic properties (Figure 9b).



(a) Row and Column norms of SOAP update $\frac{M_{\text{rot}}}{\sqrt{V_{\text{rot}}}}$ matrix in the rotated space.



(b) Row and Column norms of SOAP momentum (M_{rot}) matrix in the rotated space.



(c) Left: Distribution of the update matrix elements in the rotated space. Right: The Q-Q plot for the distribution shown on the left. The adherence to the straight line validates the Gaussianity of the elements.

Figure 9: Visualization of SOAP update statistics. The figure illustrates the underlying metrics of a gate_proj block of a 130M LLaMA model at training step 5000, capturing the mid-to-late stage dynamics.

D.2 Ablation Studies and Hyper-parameter Robustness

Ablation Studies. Under the experimental setting for the 0.25B model in Figure 4, we tested applying uniform damping coefficients $\beta_{1,2} = 0.5$ and $\beta_{1,2} = 1.0$. Despite these values being smaller than the coefficient used for flat directions in Muon-LITE-H ($\beta_2 = 2.0$), they yielded terminal losses of 2.113 and 2.129 respectively, both even inferior to the vanilla Muon baseline (2.110). The results validate our design choice of maintaining hyper-parameters along sharp directions when selectively accelerating the flat ones.

Hyper-parameter Robustness. Although LITE introduces extra hyper-parameters $(d_s, \chi, \beta_{1,2})$ to achieve acceleration, it exhibits reasonably robustness to these choices. Here we use the experimental setting for the 0.25B model in Figure 4 for illustration. Table 7 shows that Muon-LITE is insensitive to small choices of the sharp direction dimensionality. We attribute this robustness to the true sharp directions constituting a minimal fraction of the total dimensions. The top 10% subspace is adequate to cover sharp directions. Since top 10% to 20% range mainly lie in flat directions and account for a small proportion, their acceleration effect is minimal. Besides, table 8 shows that Muon-LITE achieves stable acceleration to Muon across a wide range of $(\chi, \beta_{1,2})$. Based on these findings, one

can simply adopt moderately small values for safe and stable acceleration, with further speedup attainable through extensive hyper-parameter search.

Table 7: Influence of the sharp directions dimensionality (ratio) d_s to Muon-LITE and the variants.

optimizer \ d_s	10%	20%
Muon-LITE-L	2.088	2.092
Muon-LITE-H	2.091	2.093
Muon-LITE	2.078	2.082

Table 8: Muon-LITE achieves stable loss reduction to Muon across a wide range of hyper-parameters $(\chi_{\text{adamw}}, \chi_{\text{muon}}, \beta_1, \beta_2)$, where $d_s = 0.1$ in these experiments.

configs $(\chi_{\text{adamw}}, \chi_{\text{muon}}, \beta_1, \beta_2)$	loss
$(4, 2, -0.25, 1.0)$	2.078 (-0.031)
$(4, 2, -0.25, 0.5)$	2.082 (-0.027)
$(4, 2, 0.0, 0.5)$	2.082 (-0.027)
$(2, 1, 0.0, 1.0)$	2.092 (-0.017)
$(2, 1, 0.0, 2.0)$	2.088 (-0.021)
$(2, 2, 0.0, 0.0)$	2.099 (-0.010)
$(4, 2, 0.0, 0.0)$	2.088 (-0.021)
$(1, 1, 0.0, 2.0)$	2.093 (-0.016)
$(1, 1, 0.0, 1.0)$	2.097 (-0.012)
$(1, 1, 0.0, 0.5)$	2.102 (-0.007)
$(1, 1, 0.0, 0.0)$	2.109 (Muon baseline)

D.3 Supplementary Results on Dense Models

Here we present the supplementary results for the experiments in Section 6.1.

Results of accelerating SOAP (SOAP-LITE). The results are shown in Figure 10.

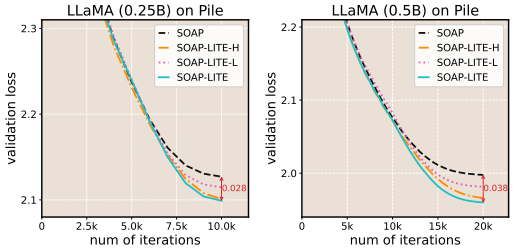


Figure 10: Performance comparison of SOAP and SOAP-LITE on LLM pre-training tasks. Same experimental setup and notations are employed as in Figure 4.

Results of downstream tasks. We evaluate the 0-shot performance of the pre-trained LLaMA 1.3B models for 25k steps (Figure 4) in Table 9. Muon-LITE demonstrates consistent improvements over the Muon baseline across the eight evaluated tasks. Notably, Muon-LITE achieves a substantial gain on BoolQ and exhibits strong generalization on reasoning-heavy benchmarks, showing significant margins on ARC-Challenge (+1.79) and MMLU (+1.39). These results indicate that MUON-LITE effectively enhances model performance across diverse domains ranging from common sense reasoning to truthful QA.

Results under different random seeds The loss variation across different seeds is negligible relative to the reduction by LITE (larger than 2e-2), consistently below 2e-3. Table 10 reports the results of Muon and Muon-LITE on the 250M LLaMA2 model across four random seeds, indicating that seed-induced loss differences are negligible.

Table 9: Evaluation results on downstream tasks (0-shot with lm-evaluation-harness) of LLaMA models (1.3B) pre-trained on Pile using Muon-LITE and Muon. The best scores in each column are bolded. Abbreviations: `T_mc2=TruthfulQA_mc2`, `AVG=Average score`.

Method	ARC_E	ARC_C	PIQA	HellaSwag	BOOLQ	WinoGrande	MMLU	T_mc2	AVG
Muon	52.99	22.10	67.41	35.30	52.45	53.35	23.76	40.14	43.44
Muon-LITE	**54.38**	**23.89**	**67.63**	**36.72**	**60.52**	**54.78**	**25.15**	**40.83**	**45.49**

Table 10: Comparison of Muon and Muon-LITE across 4 random seeds (with mean $\pm$ std)

Muon	2.10962	2.10986	2.11034	2.10927	2.10977 ± 0.00045
Muon-LITE	2.07828	2.07930	2.07979	2.07956	2.07923 ± 0.00067

E Discussions on Momentum Formulations and their Continuous-Time ODE Counterparts

E.1 Continuous-Time ODE Formulations

In this subsection, we introduce the continuous-time limits of momentum-based algorithms and demonstrate how to derive discrete optimizers from these continuous formulations (modeled as second-order or first-order systems). The derivation proceeds by reducing the second-order dynamics into a first-order system, followed by numerical discretization. This pipeline serves as a foundation for the Riemannian setting (3), which is the primary focus of this work (see Appendix G). We consider the inertial system with Hessian damping (ISHD) mentioned in Section 3.2

$$\ddot{w}_t + \alpha_t \dot{w}_t + \beta_t \nabla^2 f(w_t)\dot{w}_t + \gamma_t \nabla f(w_t) = 0, \tag{9}$$

where $\alpha_t, \beta_t, \gamma_t \geq 0$ denote the coefficients for momentum decay, Hessian damping, and the gradient driving force, respectively. By introducing an auxiliary momentum variable m_t to track the velocity $\dot{w}_t$, we can reformulate this second-order equation (9) into an equivalent first-order system:

$$\begin{cases} \dot{w}_t = -\gamma_t m_t - \beta_t \nabla f(w_t), \\ \dot{m}_t = -\alpha_t m_t + (1 - \alpha_t \beta_t/\gamma_t - \dot{\beta}_t)\nabla f(w_t). \end{cases}$$

Applying a discretization with step size $h > 0$ yields the following update rule at the k-th iteration:

$$\begin{cases} m_k = (1 - \alpha_k h)m_{k-1} + (1 - \alpha_k \beta_k/\gamma_k - \dot{\beta}_k)h\nabla f(w_k), \\ w_{k+1} = w_k - \gamma_k h m_k - \beta_k h \nabla f(w_k). \end{cases}$$

This formulation subsumes various momentum methods, recovering Heavy Ball momentum when $\beta_k = 0$ and Nesterov-type momentum when $\beta_k > 0$. Crucially, the Hessian damping term $\beta_t \nabla^2 f(w_t)\dot{w}_t$ and its discretized counterpart $\beta_k \nabla f(w_t)$ serves to enhance standard Heavy Ball momentum by suppressing oscillations. We provide a detailed discussion in Appendix E.2.

E.2 Heavy Ball Momentum, Nesterov Momentum, and Hessian Damping

In this section, we discuss equivalent formulations of Nesterov-type momentum and Heavy Ball momentum (also known as Polyak momentum). Furthermore, we analyze the acceleration dynamics of Nesterov momentum, highlighting the intrinsic connections between **Hessian damping and gradient correction.** For brevity, we assume the preconditioner is the identity matrix ($F = I$) throughout this discussion.

Heavy Ball Momentum. The Heavy Ball momentum update is defined as:

$$\begin{cases} m_k = (1 - \alpha)m_{k-1} + \nabla f(w_k), \\ w_{k+1} = w_k - \eta m_k. \end{cases}$$

This recurrence can be rewritten in a single-variable form equivalent to:

$$w_{k+1} = w_k + (1 - \alpha)(w_k - w_{k-1}) - \eta \nabla f\left(w_k\right). \tag{10}$$

This formulation aligns with the momentum schemes adopted in AdamW and many other adaptive optimizers. Note that here, the coefficient α of the gradient term (typical in EMA-style updates) is absorbed into the effective learning rate η.

Nesterov Momentum. Unfortunately, Heavy Ball momentum lacks a theoretical guarantee of globally achieving a faster convergence rate than Gradient Descent, even in deterministic strongly convex settings. In contrast, Nesterov momentum provably achieves acceleration: the convergence rates for Gradient Descent and Nesterov momentum are approximately $(1-\frac{1}{\kappa})^k$ and $(1-\frac{1}{\sqrt{\kappa}})^k$ respectively, where κ denotes the condition number.

Nesterov momentum with constant coefficients is typically formulated as:

$$\begin{cases} x_k = w_{k-1} - \eta \nabla f(w_{k-1}), \\ w_k = x_k + (1-\alpha)(x_k - x_{k-1}). \end{cases} \tag{11}$$

Eliminating the auxiliary variable x or w yields two equivalent interpretations of (11):

$$x_{k+1} = x_k + (1-\alpha)(x_k - x_{k-1}) - \eta \underbrace{\nabla f\left(x_k + (1-\alpha)(x_k - x_{k-1})\right)}_{\text{gradient correction}},$$

and

$$w_{k+1} = w_k + (1-\alpha)(w_k - w_{k-1}) - \eta \nabla f\left(w_k\right) - (1-\alpha)\eta \underbrace{\left(\nabla f(w_k) - \nabla f(w_{k-1})\right)}_{\text{gradient correction}}. \tag{12}$$

The Gradient Correction (Hessian Damping) Insight. Comparing (12) with the Heavy Ball form (10), it is evident that Nesterov momentum introduces an additional **gradient correction term**. Using a Taylor expansion, we observe that:

$$\begin{aligned} &\nabla f(x_k + (1-\alpha)(x_k - x_{k-1})) \approx (1-\alpha)\nabla^2 f(x_k)(x_k - x_{k-1}), \\ &\nabla f(w_k) - \nabla f(w_{k-1}) \approx \nabla^2 f(w_k)(w_k - w_{k-1}). \end{aligned}$$

This reveals that the gradient correction implicitly incorporates **Hessian information**. This mechanism is the key to the acceleration of Nesterov momentum: it exerts an inertial damping effect (often referred to as Hessian damping) which is proportional to the curvature. Consequently, this term applies stronger damping in directions corresponding to large Hessian eigenvalues, effectively suppressing oscillations.

The formulation in (12) is also mathematically equivalent to:

$$w_{k+1} = w_k + (1-\alpha)(w_k - w_{k-1}) - \eta_1(1+\alpha\beta)\nabla f\left(w_k\right) - \beta\eta_1(1-\alpha) \underbrace{\left(\nabla f(w_k) - \nabla f(w_{k-1})\right)}_{\text{gradient correction}}. \tag{13}$$

with $\eta_1 = \eta/(1+\alpha\beta)$ and $\beta = 1/(1-\alpha)$. Note that (13) can be transformed as:

$$\begin{cases} m_k = (1-\alpha)m_{k-1} + \nabla f(w_k), \\ w_{k+1} = w_k - \eta_1(m_k + \beta \nabla f(w_k)), \end{cases} \tag{14}$$

which corresponds to the discretized form discussed in Section 3.2.

We can observe that the term $\beta\nabla f(w_k)$ in (14) (*i*.e. the gradient correction term in (13)) arises from discretizing the Hessian damping term $\beta\nabla^2 f(w_t)\dot{w}_t$ in Eq. (1).

These insights lead to two key conclusions:

1. **Implicit Hessian Awareness:** Compared with Heavy ball momentum, Nesterov momentum introduces an additional gradient correction term that implicitly encapsulates Hessian information (*i*.e. Hessian damping), which is instrumental in achieving acceleration.
2. **Continuous-Discrete Correspondence:** This gradient correction term naturally arises from the discretization of the Hessian damping term found in the continuous-time limiting flow.

Further Improving Nesterov momentum Although Nesterov momentum uses Hessian damping (via gradient correction) to achieve acceleration, the coefficient β of this term is rigidly constrained by the momentum parameter (*i*.e. $\beta = 1/(1-\alpha)$). It is therefore natural to relax this constraint and search for an optimal β, a direction supported by recent theoretical results [3, 53]. In this work, we adopt this generalized perspective and incorporate richer second-order information, employing a larger β in flat directions than in sharp directions to further enhance acceleration.

E.3 Modeling AdEMAMix via a Third-Order ODE

The AdEMAMix optimizer [29] builds upon the paradigm of AdamW by incorporating an additional "slow" momentum term. At the k-th iteration, the parameter update rule is given by:

$$\begin{aligned} m_k^{\text{fast}} =&(1-\alpha_1)m_{k-1}^{\text{fast}} + \alpha_1 g_k, \\ m_k^{\text{slow}} =&(1-\alpha_2)m_{k-1}^{\text{slow}} + \alpha_2 g_k, \\ v_k =&(1-\beta_{\text{adam}})v_{k-1} + \beta_{\text{adam}} g_k^{\odot 2} \\ w_{k+1} =&w_k - \eta_k \frac{m_k^{\text{fast}} + \kappa m_k^{\text{slow}}}{v_k}, \end{aligned} \tag{15}$$

where g_k denotes the stochastic gradient. For brevity, we omit the weight decay and assume constant momentum hyper-parameters. Typically, α_1 is set to approximately 0.1, while α_2 takes much smaller values, such as 10^{-4}.

To derive its continuous-time version, we first simplify the analysis by omitting the preconditioner v_k and treating the learning rate η_k as a constant η. This yields the following system of differential equations:

$$\begin{aligned} \frac{d}{dt}m_t^{\text{fast}} =& -\alpha_1 m_{k-1}^{\text{fast}} + \alpha_1 \nabla f(w_k), \\ \frac{d}{dt}m_t^{\text{slow}} =& -\alpha_2 m_{k-1}^{\text{slow}} + \alpha_2 \nabla f(w_k), \\ \frac{d}{dt}w_t =& -\eta(m_t^{\text{fast}} + \kappa m_t^{\text{slow}}). \end{aligned}$$

We note that applying a discretization with step size $h = 1$ to the above system recovers the update rules in (15). Next, we compute the second and third-order time derivatives of the parameters, $\frac{d^2}{dt^2}w_t$ and $\frac{d^3}{dt^3}w_t$:

$$\begin{aligned} \frac{d^2}{dt^2}w_t =& -\eta(-\alpha_1 m_t^{\text{fast}} - \alpha_2\kappa m_t^{\text{slow}} + (\alpha_1 + \kappa\alpha_2)\nabla f(w_t)), \\ \frac{d^3}{dt^3}w_t =& -\eta(\alpha_1^2 m_t^{\text{fast}} + \alpha_2^2\kappa m_t^{\text{slow}} - (\alpha_1^2 + \kappa\alpha_2^2)\nabla f(w_t) + (\alpha_1 + \kappa\alpha_2)\nabla^2 f(w_t)\frac{d}{dt}w_t). \end{aligned} \tag{16}$$

By combining (16) with the velocity equation $\frac{d}{dt}w_t = -\eta(m_t^{\text{fast}} + \kappa m_t^{\text{slow}})$ to eliminate the momentum variables m_t^{fast} and m_t^{slow}, we arrive at the following third-order ODE :

$$\underbrace{\frac{d^3}{dt^3}w_t}_{\text{jerk}} + (\alpha_1+\alpha_2)\underbrace{\frac{d^2}{dt^2}w_t}_{\substack{\text{accelerated}\\ \text{velocity}}} + \alpha_1\alpha_2\underbrace{\frac{d}{dt}w_t}_{\text{velocity}} + \eta(\alpha_1+\kappa\alpha_2)\underbrace{\nabla^2 f(w_t)\frac{d}{dt}w_t}_{\text{Hessian damping}} + \eta(1+\kappa)\alpha_1\alpha_2\underbrace{\nabla f(w_t)}_{\text{gradient}} = 0. \tag{17}$$

Similar to (3), the Riemannian counterpart of (17) can be derived as

$$\begin{aligned} &\nabla_{\dot{w}_t}\nabla_{\dot{w}_t}\dot{w}_t + (\alpha_1+\alpha_2)\nabla_{\dot{w}_t}\dot{w}_t + \alpha_1\alpha_2\frac{d}{dt}w_t \\ &\qquad + \eta(\alpha_1+\kappa\alpha_2)\frac{d}{dt}\operatorname{grad} f(w_t) + \eta(1+\kappa)\alpha_1\alpha_2 \operatorname{grad} f(w_t) = 0, \end{aligned}$$

where we use $\operatorname{Hess}(w_t)\dot{w}_t = \frac{d}{dt}\operatorname{grad} f(w_t)$.

Intuitive Interpretation. Standard momentum methods (like Heavy Ball) typically correspond to second-order ODEs, describing a particle with mass and friction. In contrast, Eq. (17) reveals that AdEMAMix is governed by a **third-order** dynamics. The term $\frac{d^3}{dt^3}w_t$ introduces the concept of "jerk" (rate of change of accelerated velocity $\ddot{w}_t$) into the optimization trajectory. The third-order structure with Hessian damping allows the optimizer to regulate not only velocity but also its rate of change across multiple time scales. As a result, transient gradient fluctuations induced by rapidly varying curvature are attenuated before influencing parameter updates, enabling more stable and robust navigation of highly ill-conditioned landscapes.

F Foundations of Riemannian Geometry

In this section, we provide a detailed introduction to the foundation for Riemannian Geometry of $(\mathbb{R}^p, F)$ considered in the main text. The parameter space is the Euclidean space $\mathbb{R}^p$, while the metric is the Riemannian metric induced by the preconditioner $F(w)$, which is formally denoted as the manifold $\mathcal{M} = (\mathbb{R}^p, F)$.

F.1 Riemannian Metric

The core structure of our framework is based on Riemannian metric, a smooth matrix-value map $F : \mathcal{M} = \mathbb{R}^p \to \mathbb{S}^p_{++}$ that assigns an inner product to the tangent space $T_w\mathcal{M} \cong \mathbb{R}^p$ at each point $w \in \mathcal{M}$. For any tangent vectors $u, v \in T_w\mathcal{M}$, the inner product is defined as $\langle u, v\rangle_{F(w)} \coloneqq u^\top F(w)v$. This metric induces a norm $\|u\|_{F(w)} = \sqrt{\langle u, u\rangle_{F(w)}}$ in the tangent space $T_w\mathcal{M}$, which measures the magnitude of perturbations u relative to the local curvature $F(w)$.

While the tangent space $T_w\mathcal{M}$ characterizes velocities or perturbations, its dual, the cotangent space $T^*_w\mathcal{M}$, consists of differentials that act linearly on these velocities, naturally representing gradients of functions. Formally, $T^*_w\mathcal{M}$ is the dual space of $T_w\mathcal{M}$, consisting of all linear functionals $\xi : T_w\mathcal{M} \to \mathbb{R}$. It also holds that $T_w\mathcal{M} \cong \mathbb{R}^p$. The cotangent space is equipped with the dual norm $\|u\|_{F(w)^{-1}} = \sqrt{u^\top F(w)^{-1}u}$ for $u \in T^*\mathcal{M}$. In particular, the Riemannian metric $F(w)$ induces a natural isomorphism between the tangent and cotangent spaces:

$$F(w) : T_w\mathcal{M} \to T^*_w\mathcal{M}, \quad F(w)^{-1} : T^*_w\mathcal{M} \to T_w\mathcal{M}.$$

Under this identification, an element $\xi \in T^*_w\mathcal{M}$ acts as a linear functional on the tangent space via the Euclidean inner product $\xi(u) = \xi^\top u$ for $u \in T_w\mathcal{M}$. Inner-product structures of $T_w\mathcal{M}, T^*_w\mathcal{M}$ and between them are $\langle\cdot,\cdot\rangle_{F(w)}$, $\langle\cdot,\cdot\rangle_{F(w)^{-1}}$, $\langle\cdot,\cdot\rangle$ respectively.

Based on the Riemannian metric, we can define the length of a smooth curve $\gamma : [0, 1] \to \mathcal{M}$. The Riemannian length $L(\gamma)$ is given by integrating the local norm of the velocity vector $\dot\gamma(t)$:

$$L(\gamma) = \int_0^1 \|\dot\gamma(t)\|_{F(\gamma(t))}dt = \int_0^1 \sqrt{\dot\gamma(t)^\top F(\gamma(t))\dot\gamma(t)}dt.$$

The Riemannian distance (or geodesic distance) between two points $x, y \in \mathcal{M} = \mathbb{R}^p$ is defined as the infimum of lengths over all piecewise smooth curves connecting them:

$$d_F(x, y) = \inf_{\gamma\in D} L(\gamma),$$

where D is the set containing all smooth curves γ connecting (x, y) with $\gamma(0) = x, \gamma(1) = y$.

The distance reduces to the Euclidean metric when $F = I$. Furthermore, when (x, y) are sufficiently close locally, we have $d_F(x, y) \approx \|x - y\|_{F(x)}$. The Riemannian gradient $\operatorname{grad} f$ is an example of vector field.

Smooth Vector Fields and the Tangent Bundle. With the tangent space $T_w\mathcal{M}$ defined at each point, we consider the collection of all tangent vectors across the manifold, known as the tangent bundle, denoted by $T\mathcal{M} \coloneqq \bigcup_{w\in\mathcal{M}}(\{w\} \times T_w\mathcal{M})$. The tangent bundle is itself a smooth manifold of dimension $2p$. A vector field V is a mapping (formally called a *section*) that assigns to every point $w \in \mathcal{M}$ a specific tangent vector $V(w) \in T_w\mathcal{M}$. The space of smooth vector fields, denoted by $\mathcal{C}^\infty(\mathcal{M}, T\mathcal{M})$, consists of all such fields V where the assignment varies smoothly with w. In the global coordinate system of $\mathcal{M} \cong \mathbb{R}^p$, any vector field $V \in \mathcal{C}^\infty(\mathcal{M}, T\mathcal{M})$ can be uniquely expressed as a linear combination of the coordinate basis vectors $\{\frac{\partial}{\partial w_1}, \ldots, \frac{\partial}{\partial w_p}\}$ (which form the standard basis for each $T_w\mathcal{M}$):

$$V(w) = \sum_{i=1}^p v_i(w)\frac{\partial}{\partial w_i},$$

where each component function $v_i : \mathcal{M} \to \mathbb{R}$ is a smooth scalar function. For a function f, $(Vf)|_w \coloneqq V(w)f(w) = \sum_{i=1}^p v_i(w)\frac{\partial}{\partial w_i}f(w)$ and $VUf \coloneqq V(Uf)$ for vector field U.

F.2 The Levi-Civita Connection

In Euclidean spaces, comparing vectors at different points is trivial because the tangent spaces are identical. On a curved manifold, tangent spaces $T_x\mathcal{M}$ and $T_y\mathcal{M}$ are distinct inner-product spaces. To differentiate vector fields, we require an affine connection ∇, which provides a rule for connecting adjacent tangent spaces.

Definition. The Levi-Civita connection $\nabla_{(\cdot)}(\cdot) : \mathcal{C}^\infty(\mathcal{M}, T\mathcal{M}) \times \mathcal{C}^\infty(\mathcal{M}, T\mathcal{M}) \to \mathcal{C}^\infty(\mathcal{M}, T\mathcal{M})$ is the fundamental connection associated with the metric F. It is uniquely determined by two geometric conditions:

1. Torsion-Free: For any vector fields $X, Y \in \mathcal{C}^\infty(\mathcal{M}, T\mathcal{M})$, the connection is symmetric: $\nabla_X Y - \nabla_Y X = [X, Y]$, where $[X, Y]$ is the Lie bracket (in $\mathbb{R}^p$, defined by $[X, Y] = XY - YX$).
2. Metric Compatibility: The connection preserves the inner product structure. Formally, for vector fields X, Y, Z:

$$X\langle Y, Z\rangle_F = \langle \nabla_X Y, Z\rangle_F + \langle Y, \nabla_X Z\rangle_F.$$

Coordinate Representation (Christoffel Symbols). Since our manifold is globally $\mathbb{R}^p$, we can express the covariant derivative explicitly using the standard basis $\{e_1, \ldots, e_p\}$. Let $\Gamma_{ij}^k(w)$ denote the Christoffel symbols of the second kind. The covariant derivative of a vector field v along u is given component-wise by:

$$(\nabla_u v)_k = \sum_{i=1}^{p} u_i \frac{\partial v_k}{\partial w_i} + \sum_{i,j \le p} \Gamma_{ij}^k(w) u_i v_j.$$

The first term captures the standard Euclidean directional derivative, while the second term corrects for the twisting of the coordinate system induced by the metric. The Christoffel symbols are derived from the metric $F(w)$:

$$\Gamma_{ij}^k = \frac{1}{2} \sum_l (F^{-1})_{kl} \left(\frac{\partial F_{jl}}{\partial w_i} + \frac{\partial F_{il}}{\partial w_j} - \frac{\partial F_{ij}}{\partial w_l} \right).$$

Here, $(F^{-1})_{kl}$ denotes the (k, l)-entry of the inverse metric matrix, and indices denote partial derivatives with respect to coordinates. The Levi-connection would reduce to the Euclidean directional derivative when $F = I$, as $\Gamma_{ij}^k = 0$ in this case.

F.3 Riemannian Gradient and Hessian

With the metric and connection defined, we can rigorously define the following differential operators.

Riemannian Gradient. The Riemannian gradient $\text{grad} f(w)$ is the unique tangent vector representing the differential $df(w)$ via the metric:

$$\langle \text{grad} f(w), u\rangle_{F(w)} = df(w)[u] := \lim_{\epsilon \to 0} \frac{f(w + \epsilon u) - f(w)}{\epsilon} \text{ for any } u \in \mathbb{R}^p.$$

It has the closed form

$$\text{grad} f(w) = F(w)^{-1} \nabla f(w),$$

where $\nabla f(w)$ is the standard Euclidean gradient. This confirms that the preconditioner F^{-1} acts as the "inverse metric" mapping cotangent vectors (gradients) to tangent vectors (directions).

Riemannian Hessian. The Riemannian Hessian is a linear operator on the tangent space, defined as the covariant derivative of the gradient vector field:

$$\text{Hess} f(w)[u] = \nabla_u \text{grad} f(w), \quad \forall u \in T_w\mathcal{M}.$$

Unlike the Euclidean Hessian $\nabla^2 f(w)$, the Riemannian Hessian accounts for the curvature of the manifold. Explicitly:

$$\text{Hess} f(w) = F(w)^{-1} \left(\nabla^2 f(w) - \mathcal{K}(w) \right),$$

where

$$[\mathcal{K}(w)]_{ij} = \sum_{k=1}^{p} \Gamma_{ij}^{k}(w) \frac{\partial f}{\partial w_k}(w).$$

When $F(w)$ changes slowly with w, we have $\Gamma_{ij}^{k} \approx 0$ and $\operatorname{Hess} f(w) \approx F(w)^{-1} \nabla^2 f(w)$.

G Transforming the Second Order RISHD into a First Order System

Lemma 1. *Let $\mathcal{M} = \mathbb{R}^p$ be endowed with a Riemannian metric tensor $F(w) \succ 0$. For any vector fields $u, v \in \mathcal{C}^\infty(\mathcal{M}, T\mathcal{M})$, we have the following expression for the Levi-Civita connection:*

$$\nabla_u v = \underbrace{\partial_u v}_{\substack{\text{Euclidean} \\ \text{derivative}}} + \underbrace{\frac{1}{2} F^{-1} \left[(\partial_u F) v + (\partial_v F) u - \nabla_w (u^\top F v) \right]}_{\text{Geometric correction}}.$$

Proof. By Koszul formula (Corollary 5.11 in [19]), we have

$$2 \langle \nabla_u v, h \rangle_F = \partial_u \langle v, h \rangle_F + \partial_v \langle u, h \rangle_F - \partial_h \langle u, v \rangle_F + \langle [u, v], h \rangle_F - \langle [u, h], v \rangle_F - \langle [v, h], u \rangle_F. \tag{18}$$

Substituting

$$\begin{aligned}
\partial_u (v^\top F h) &= (\partial_u v)^\top F h + v^\top (\partial_u F) h + v^\top F (\partial_u h), \\
\partial_v (u^\top F h) &= (\partial_v u)^\top F h + u^\top (\partial_v F) h + u^\top F (\partial_v h), \\
-\partial_h (u^\top F v) &= -(\partial_h u)^\top F v - u^\top (\partial_h F) v - u^\top F (\partial_h v), \\
[u, v] &= \partial_u v - \partial_v u, \\
[u, h] &= \partial_u h - \partial_h u, \\
[v, h] &= \partial_v h - \partial_h v,
\end{aligned}$$

into (18) yields

$$2 h^\top F \nabla_u v = h^\top \left[2 F \partial_u v + (\partial_u F) v + (\partial_v F) u - \nabla_w (u^\top F v) \right].$$

We get the conclusion.

□

Proposition 2. *Let $\mathcal{M} = \mathbb{R}^d$ be endowed with a Riemannian metric tensor $F(w) \succ 0$. Then the RISHD*

$$\nabla_{\dot{w}_t} \dot{w}_t + \alpha_t \dot{w}_t + \beta_t \nabla_{\dot{w}_t} \operatorname{grad} f(w_t) + \gamma_t \operatorname{grad} f(w_t) = 0. \tag{19}$$

with $\alpha_t = \alpha - \dot{\eta}_t / \eta_t$, $\beta_t = \beta \eta_t$, $\gamma_t = \eta_t (\alpha \beta + 1)$ is equivalent to the following first order system:

$$\begin{cases} \dot{w}_t = -\eta_t F(w_t)^{-1} (m_t + \beta \nabla f(w_t)), \\ \dot{m}_t = -\alpha m_t + \nabla f(w_t) + R_t, \end{cases}$$

where we define $u_t = m_t + \beta \nabla f(w_t)$ for notational brevity and

$$\begin{aligned}
R_t = &\frac{\eta_t}{2} \nabla_w (u_t^\top F^{-1}(w) m_t)|_{w=w_t} + \frac{\beta \eta_t}{2} \nabla F(w_t)[F^{-1}(w_t) u_t] F^{-1}(w_t) \nabla f(w_t) \\
&- \frac{\beta \eta_t}{2} \nabla F(w_t)[F^{-1}(w_t) \nabla f(w_t)] F^{-1}(w_t) u_t.
\end{aligned} \tag{20}$$

Remark 4. (19) *is equivalent to* (3) *as* $\nabla_{\dot{w}_t} \operatorname{grad} f(w_t) = \operatorname{Hess} f(w_t)[\dot{w}_t]$.

Proof. First, we let $m_t = -F(w_t) \dot{w}_t / \eta_t - \beta \nabla f(w_t)$ and get

$$\begin{aligned}
\dot{w}_t = &- \eta_t F(w_t)^{-1} (m_t + \beta \nabla f(w_t)), \\
\ddot{w}_t = &- \dot{\eta}_t F(w_t)^{-1} (m_t + \beta \nabla f(w_t)) - \eta_t \frac{d}{dt} F(w_t)^{-1} (m_t + \beta \nabla f(w_t)) \\
&- \eta_t F(w_t)^{-1} (\dot{m}_t + \beta \nabla^2 f(w_t) \dot{w}_t).
\end{aligned}$$

By Lemma 1, we have

$$\nabla_{\dot{w}_t}\dot{w}_t = \ddot{w}_t + F(w_t)^{-1}\dot{F}(w_t)\dot{w}_t - \frac{1}{2}F^{-1}(w_t)\nabla_w(\dot{w}_t^\top F(w)\dot{w}_t)|_{w=w_t},$$

and

$$\begin{aligned}
&\nabla_{\dot{w}_t}\operatorname{grad} f(w_t)\\
=&\nabla_{\dot{w}_t}(F^{-1}(w_t)\nabla f(w_t))\\
=&\frac{d}{dt}(F^{-1}(w_t))\nabla f(w_t) + F^{-1}(w_t)\nabla^2 f(w_t)\dot{w}_t + \frac{1}{2}F^{-1}(w_t)\frac{d}{dt}F(w_t)F^{-1}(w_t)\nabla f(w_t)\\
&+\frac{1}{2}F^{-1}(w_t)\nabla F(w_t)[F^{-1}(w_t)\nabla f(w_t)]\dot{w}_t - \frac{1}{2}F^{-1}(w_t)\nabla_w(\dot{w}_t^\top F(w)F^{-1}(w_t)\nabla f(w_t)|_{w=w_t}\\
=&F^{-1}(w_t)\nabla^2 f(w_t)\dot{w}_t - \frac{1}{2}F^{-1}(w_t)\frac{d}{dt}F(w_t)F^{-1}(w_t)\nabla f(w_t)\\
&+\frac{1}{2}F^{-1}(w_t)\nabla F(w_t)[F^{-1}(w_t)\nabla f(w_t)]\dot{w}_t - \frac{1}{2}F^{-1}(w_t)\nabla_w(\dot{w}_t^\top F(w)F^{-1}(w_t)\nabla f(w_t))|_{w=w_t}.
\end{aligned}$$

Substituting them into (19) and using $\nabla_w(\dot{w}_t^\top F(w)\dot{w}_t) = -\eta_t^2\nabla_w(u_t^\top F^{-1}(w)u_t)$ yields

$$\begin{aligned}
\dot{m}_t =& -\alpha m_t + \nabla f(w_t) + \frac{\eta_t}{2}\nabla_w(u_t^\top F^{-1}(w)u_t)\big|_{w=w_t}\\
&-\frac{\beta}{2}\nabla F(w_t)[\dot{w}_t]F^{-1}(w_t)\nabla f(w_t) + \frac{\beta}{2}\nabla F(w_t)[F^{-1}(w_t)\nabla f(w_t)]\dot{w}_t\\
&-\frac{\beta}{2}\nabla_w\left(\dot{w}_t^\top F(w)F^{-1}(w_t)\nabla f(w_t)\right)\big|_{w=w_t}.
\end{aligned}$$

We get the conclusion. □

Remark 5. *In general, we assume that $F(w)$ varies slowly with respect to w to ensure the stability of the preconditioner. This assumption is practically justified by the common use of Exponential Moving Average (EMA) with a decay coefficient close to 1, which enforces a smooth evolution of the estimated preconditioner. Consequently, based on the expression of R_t in (20), this term is typically negligible relative to m_t and $\nabla f(w_t)$, as it scales with the small step size η_t and the limited variation $\|\nabla F(w)\|_{\mathrm{op}}$.*

H Theoretical Analysis on Quadratic Objectives: An Illustrative Example

To demonstrate the acceleration mechanism of LITE, we analyze the discrete-time dynamics of the update rule (2), which corresponds to non-accelerated adaptive optimizers, on a quadratic objective function:

$$f(w) = \frac{1}{2}\sum_{i=1}^{p}\lambda_i w_i^2 + b^\top w.$$

We assume the eigenvalues are sorted as $\lambda_1 > \cdots > \lambda_k > 0 \geq \cdots \geq \lambda_p$, representing an anisotropic landscape where negative λ_i correspond to local nonconvex directions. For simplicity, we set the step size $h = 1$ and the preconditioner $F(w) = I_p$. Since the Hessian is diagonal, the dynamics decouple across coordinates.

Let $w_\star = -[b_1/\lambda_1, \ldots, b_p/\lambda_p]^\top$ denote the stationary point. To characterize the trajectory of the error term $e_k = w_k - w_\star$, we eliminate the auxiliary momentum variable m_k. From (2) with $F = I_p$, the i-th coordinate of m_k can be expressed in terms of e_k as:

$$\eta(m_k)_i = (1 - \eta\beta\lambda_i)(e_k)_i - (e_{k+1})_i, \quad 1 \leq i \leq p. \tag{21}$$

Applying a time shift to (21), we obtain:

$$\eta(m_{k-1})_i = (1 - \eta\beta\lambda_i)(e_{k-1})_i - (e_k)_i.$$

Combining these relations leads to a homogeneous second-order linear recurrence:

$$(e_{k+1})_i - 2T_i(e_k)_i + D_i(e_{k-1})_i = 0,$$

where the coefficients are defined as:

$$T_i = T(\lambda_i) := 1 - \frac{\alpha + \eta\lambda_i(\beta+1)}{2}, \quad D_i = D(\lambda_i) := (1-\alpha)(1-\eta\beta\lambda_i).$$

The characteristic equation is given by $r^2 - 2T_i r + D_i = 0$, with roots:

$$r_{i,1,2} = T_i \pm \sqrt{T_i^2 - D_i}.$$

Based on the discriminant $\Delta_i = T_i^2 - D_i$, the dynamics fall into three regimes, yielding the following closed-form solutions ($c_{i,1}, c_{i,2}$ are constants determined by initialization):

$$w_{k,i} = -\frac{b_i}{\lambda_i} + \begin{cases} c_{i,1} r_{i,1}^k + c_{i,2} r_{i,2}^k, & \Delta_i > 0 \text{ (\textbf{Overdamped})}, \\ (c_{i,1} + c_{i,2}k) D_i^{k/2}, & \Delta_i = 0 \text{ (\textbf{Critically damped})}, \\ D_i^{k/2}\left[c_{i,1}\cos(k\theta_i) + c_{i,2}\sin(k\theta_i)\right], & \Delta_i < 0 \text{ (\textbf{Underdamped})}, \end{cases}$$

where $\theta_i = \arccos(T_i/\sqrt{D_i})$.

Stability Analysis. Convergence in convex directions requires the magnitude of all characteristic roots to be at most 1. According to the Jury Stability Criterion, the necessary and sufficient conditions are $|P_i(0)| \le 1$, $P_i(1) \ge 0$, and $P_i(-1) \ge 0$, where $P_i(z) = z^2 - 2T_i z + D_i$. Solving these inequalities yields the stability bound:

$$\eta \le \frac{2(2-\alpha)}{\lambda_1 \max\{1 + 2\beta - \alpha\beta,\, 2\beta(1-\alpha)\}}. \tag{22}$$

Ideally, this condition confirms that the maximum allowable step size is governed by the sharpest direction λ_1.

Regime Classification. We next analyze how the curvature λ_i determines the dynamic regime. The equation $T(\lambda)^2 = D(\lambda)$ with respect to λ:

$$\left(1 - \frac{\alpha + \eta\lambda(\beta+1)}{2}\right)^2 = (1-\alpha)(1-\eta\beta\lambda)$$

possesses two real positive roots, denoted $\tilde{\lambda}_1 < \tilde{\lambda}_2$. By the properties of quadratic functions, we refer eigenvalues λ satisfying $\lambda < \tilde{\lambda}_1$ to *flat directions (parts)*, which fall into the **overdamped regime**. Conversely, eigenvalues $\lambda \ge \tilde{\lambda}_1$ correspond to *sharp directions*, whose dynamic regimes depend on the specific magnitude of the curvature (typically transitioning into the underdamped regime).

Acceleration Mechanism of LITE**.** Here we suppose that the stability condition (22) holds. Since LITE explicitly increases η and β in flat directions, we focus our analysis on the regime $\lambda_i < \tilde{\lambda}_1$. In this regime, the dynamics are overdamped, and dominated by the larger root $r_1 = T + \sqrt{T^2 - D}$. We demonstrate that our parameter adjustment modifies r_1 to facilitate escape from nonconvex regions and foster convergence in convex regions.

We analyze the sensitivity of r_1 to β and η case by case:

- **Case 1: Flat Nonconvex Directions ($\lambda < 0$).** The coefficients become $T = 1 - \frac{\alpha}{2} + \frac{\eta|\lambda|(\beta+1)}{2}$ and $D = (1-\alpha)(1+\eta\beta|\lambda|)$. Clearly, $T > 1 - \alpha$, and both T and D are strictly increasing functions of β and η. Consider the partial derivatives:

  $$\frac{\partial r_1}{\partial \beta} = \frac{\partial T}{\partial \beta} + \frac{1}{2\sqrt{T^2 - D}}\left(2T\frac{\partial T}{\partial \beta} - \frac{\partial D}{\partial \beta}\right).$$

  Since $T > 0$, and noting that:

  $$2T\frac{\partial T}{\partial \beta} - \frac{\partial D}{\partial \beta} = T\eta|\lambda| - (1-\alpha)\eta|\lambda| > 0,$$

  (and similarly for η), we conclude that increasing β or η yields a strictly larger r_1 (with $r_1 > 1$). This increases the exponential divergence rate along negative curvature directions, thereby facilitating rapid escape.

- **Case 2: Flat Convex Directions ($0 < \lambda < \tilde{\lambda}_1$).** In this regime, the convergence rate is governed by how quickly r_1 decays to zero. To accelerate convergence, we must **minimize** r_1. The coefficients are $T = 1 - \frac{\alpha+\eta\lambda(\beta+1)}{2}$ and $D = (1-\alpha)(1-\eta\beta\lambda)$. To determine the effect of β on r_1, we differentiate the characteristic equation $r_1^2 - 2Tr_1 + D = 0$ implicitly with respect to β:

$$2r_1\frac{\partial r_1}{\partial \beta} - 2\frac{\partial T}{\partial \beta}r_1 - 2T\frac{\partial r_1}{\partial \beta} + \frac{\partial D}{\partial \beta} = 0, \; implying \quad \frac{\partial r_1}{\partial \beta} = \frac{2r_1\frac{\partial T}{\partial \beta} - \frac{\partial D}{\partial \beta}}{2(r_1 - T)}.$$

Substituting $\frac{\partial T}{\partial \beta} = -\frac{\eta\lambda}{2}$ and $\frac{\partial D}{\partial \beta} = -(1-\alpha)\eta\lambda$ gives

$$\frac{\partial r_1}{\partial \beta} = \frac{\eta\lambda\big((1-\alpha) - r_1\big)}{2\sqrt{T^2 - D}}.$$

Since $\lambda > 0$, the sign of $\frac{\partial r_1}{\partial \beta}$ is opposite to the sign of $r_1 - (1-\alpha)$. Evaluating the characteristic polynomial at $1-\alpha$ yields $(1-\alpha)^2 - 2T(1-\alpha) + D = (1-\alpha)\eta\lambda > 0$. Because the leading coefficient is positive, the two roots lie on the same side of $1-\alpha$; their average is $\frac{r_1+r_2}{2} = T$. Hence $r_1 > 1-\alpha$ if $T > 1-\alpha$. The condition $T > 1-\alpha$ reduces to $\eta\lambda(\beta+1) < \alpha$, which is naturally satisfied for sufficiently small λ (i.e., in flat directions) under the stability bound (22). Consequently, we have $r_1 > 1-\alpha$, implying $\frac{\partial r_1}{\partial \beta} < 0$. Thus, increasing β strictly decreases the dominant root, directly accelerating convergence along these flat convex directions.

We now verify the analogous property for η. Similar to β we have:

$$\frac{\partial r_1}{\partial \eta} = \frac{2r_1\frac{\partial T}{\partial \eta} - \frac{\partial D}{\partial \eta}}{2(r_1 - T)}.$$

Using $\frac{\partial T}{\partial \eta} = -\frac{\lambda(\beta+1)}{2}$ and $\frac{\partial D}{\partial \eta} = -(1-\alpha)\beta\lambda$, the numerator simplifies to $\lambda\big[(1-\alpha)\beta - r_1(\beta+1)\big]$, yielding

$$\frac{\partial r_1}{\partial \eta} = \frac{\lambda\big[(1-\alpha)\beta - r_1(\beta+1)\big]}{2\sqrt{T^2 - D}}.$$

Since $r_1 > 1-\alpha$ and $\beta > 0$, we have

$$r_1 > 1-\alpha > \frac{\beta}{\beta+1}(1-\alpha),$$

which implies the numerator is strictly negative, hence $\frac{\partial r_1}{\partial \eta} < 0$. Thus, increasing β or η strictly decreases the dominant root r_1, directly accelerating convergence along these flat convex directions.

I Some Common Adaptive Optimizers for LLM Pre-training

Here we list some common optimizers in LLM pre-training. They can be subsumed by discretizing the ODE framework (2). We omit gradient clipping and AdamW-type bias correction coefficient $\frac{1-\theta^t}{\sqrt{1-\beta_v^t}}$ for notational brevity.

Algorithm 2 AdamW

Require: Hyper-parameters: θ, β_v, ϵ, $\{\eta_t\}$, λ
  Initialize $m = 0$, $v = 0$, $t = 0$
  for $t = 1, 2, \ldots$ **do**
    Compute stochastic gradient g_t at current parameters w_t.
    $m_t = \theta m_{t-1} + (1-\theta)g_t$
    $v_t = \beta_v v_{t-1} + (1-\beta_v)g_t^{\odot 2}$
    $w_{t+1} = w_t - \eta_t\left(\frac{m_t}{\sqrt{v_t}+\epsilon} + \lambda w_t\right)$
  end for

Algorithm 3 Lion

Require: $\theta, \beta_v, \{\eta_t\}, \epsilon, \lambda$
  Initialize $m = 0$
  for $t = 1, 2, \ldots$ **do**
    Compute gradient g_t at current parameters w_t
    $u_t = \theta m_{t-1} + (1-\theta) g_t$
    $m_t = \beta_v m_{t-1} + (1-\beta_v) g_t$
    $w_{t+1} = w_t - \eta_t \operatorname{sign}(u_t) - \eta_t \lambda w_t$
  end for

Algorithm 4 MARS

Require: $\theta, \beta_v, \gamma, \epsilon, \{\eta_t\}, \lambda$
  Initialize $m = 0$, $v = 0$, $g_0 = 0$
  for $t = 1, 2, \ldots$ **do**
    Compute stochastic gradient g_t at current parameters w_t
    $c_t = g_t + \gamma \dfrac{\theta}{1-\theta}(g_t - g_{t-1})$
    $m_t = \theta m_{t-1} + (1-\theta) c_t$
    $v_t = \beta_v v_{t-1} + (1-\beta_v) c_t^{\odot 2}$
    $w_{t+1} = w_t - \eta_t \dfrac{m_t}{\sqrt{v_t} + \epsilon} - \eta_t \lambda w_t$
  end for

Remark 6. *In MARS, the momentum term is updated as:*

$$m_t = \theta m_{t-1} + (1-\theta) g_t + \gamma \theta (g_t - g_{t-1}). \tag{23}$$

It can be transformed to the formulation in (2) as:

$$s_t = \theta s_{t-1} + g_t, \quad \tilde{m}_t = s_t + \frac{\gamma}{(1-\gamma)(1-\theta)} g_t,$$

with $m_t = (1-\gamma)(1-\theta)\tilde{m}_t$. Note that $\gamma \neq 1$; otherwise, (23) would reduce to $m_t = \theta m_{t-1} - \theta g_{t-1}$, thereby completely discarding the current gradient information g_t.

Algorithm 5 Muon

Require: $\theta_{\text{muon}}, \eta, \epsilon, \theta_{\text{adamw}}, \beta_v, \{\eta_t\}, \lambda$
  Initialize state m
  for $t = 1, 2, \ldots$ **do**
    For weights in Output, Embedding, and Norm layers:
      Update using AdamW with hyper-parameters θ_{adamw}, β_v, ϵ, η_t, λ.
    For weight matrices in transformer layers:
      Compute stochastic gradient g_t at current parameters w_t
      $m_t = \theta_{\text{muon}} m_{t-1} + g_t$
      $u_t = \theta_{\text{muon}} m_t + g_t$
      $w_{t+1} = w_t - \text{scale} \cdot \eta_t \operatorname{NS}(u_t) - \eta_t \lambda w_t$, where $\text{scale} = 0.2\sqrt{\max\{m, n\}}$ for $w \in \mathbb{R}^{m \times n}$.
  end for

Algorithm 6 SOAP

Require: $\theta, \beta_v, \beta_{\text{shampoo}}, k, \epsilon, \lambda$
  Initialize state: $m = 0$, $v = 0$, $L = 0$, $R = 0$, Q_l, Q_r
  for $t = 1, 2, \ldots$ **do**
    For weights in Output, Embedding, and Norm layers:
      Update using AdamW with hyper-parameters β_1, β_2, ϵ, η_t, λ.
    for weight matrices in transformer layers: do
      Compute stochastic gradient G_t at current parameters W_t
      $G_t^{\text{rot}} = Q_l^\top G_t Q_r$
      $M_t = \theta M_{t-1} + (1-\theta) G_t$
      $V_t = \beta_v V_{t-1} + (1-\beta_v)(G_t^{\text{rot}})^{\odot 2}$
      $W_{t+1} = W_t - \eta_t Q_l \left(\frac{Q_l^\top M_t Q_r}{\sqrt{V_t} + \epsilon} \right) Q_r^\top - \eta_t \lambda W_t$
      $L_t = \beta_{\text{shampoo}} L_{t-1} + (1-\beta_{\text{shampoo}}) G_t G_t^\top$
      $R_t = \beta_{\text{shampoo}} R_{t-1} + (1-\beta_{\text{shampoo}}) G_t^\top G_t$
      if $t \bmod k = 0$ **then**
        $Q_l \leftarrow \text{QR}(L_t Q_l)$ {QR decomposition}
        $Q_r \leftarrow \text{QR}(R_t Q_r)$
      end if
    end for
  end for

I.1 Preconditioner-Estimated Riemannian Metrics for Common Adaptive Optimizers

In this section, we analyze the implicit Riemannian metric structure F estimated by the practical preconditioner for each parameter block in Section 4.1. Establishing these explicit, deterministic metric forms relies on certain heuristic approximations. To make the analysis tractable, we operate under the assumptions that the stochastic gradients are heavily noise-dominated and that the underlying noise covariance $F(w)$ varies slowly with respect to the parameters w.

Let $G \in \mathbb{R}^{m \times n}$ denote the stochastic gradient in matrix form, and let $g = \text{vec}(G) \in \mathbb{R}^{mn}$ be its vectorized counterpart. A prevalent choice in the literature is to estimate the whitening metric $\hat{F}(w) = (\mathbb{E}[gg^\top])^{1/2} \in \mathbb{R}^{mn \times mn}$ [55, 8]. This formulation has demonstrated remarkable efficacy in neural network training and exhibits a close theoretical connection to the Gauss-Newton approximation of the Hessian. There are also alternative methods that explicitly estimate the Hessian matrix. Building on these premises, we demonstrate below how the practical preconditioners F_k employed in common adaptive optimizers can be viewed as approximations of their corresponding deterministic metric forms $F(w_k)$. Throughout this section, we frequently utilize the Kronecker product identity $\text{vec}(ABC) = (C^\top \otimes A)\,\text{vec}(B)$ to bridge the matrix and vector formulations.

EMA-based Preconditioners. AdamW employs an Exponential Moving Average (EMA) to update its preconditioner F_k^{-1} via $F_k = \operatorname{diag} v_k^{1/2}$ with $v_k = \beta_v v_{k-1} + (1-\beta_v) g_k \odot g_k$. This expands to $v_k = (1-\beta_v)\sum_{i=0}^{k} \beta_v^{k-i} g_i \odot g_i$, where β_v is typically close to 1. Observing that $(1-\beta_v)\sum_{i=k-t}^{k} \beta_v^{k-i} \approx 1$ for a large window t, and assuming that the stochastic gradients g_i are approximately independent samples while $\hat{F}(w)$ changes slowly with respect to w, we can deduce that $v_k^{1/2}$ provides a stable and efficient approximation of $F^{\text{adam}}(w) = \operatorname{diag} \hat{F}(w)$. This approximation logic can directly extend to other optimizers that accumulate preconditioners via EMA, such as SOAP. Specifically, for SOAP, from Algorithm 6 we have

$$F_k = (Q_r \otimes Q_l)\,(\operatorname{diag} \text{vec}(V_k))^{\frac{1}{2}}\,(Q_r \otimes Q_l)^\top, \text{ with } V_k = \beta_v V_{k-1} + (1-\beta_v)(G_k^{\text{rot}})^{\odot 2},$$

where Q_l, Q_r are updated in a lazy fashion, and $G_k^{\text{rot}} = Q_l^\top G_k Q_r$. This gives

$$F^{\text{soap}}(w) = (Q_r \otimes Q_l)\left(\operatorname{diag} \mathbb{E}[g_{\text{rot}} g_{\text{rot}}^\top]\right)^{\frac{1}{2}} (Q_r \otimes Q_l)^\top,$$

where Q_l and Q_r are the eigenvectors of $\mathbb{E}[GG^\top]$ and $\mathbb{E}[G^\top G]$, respectively, and $g_{\text{rot}} = \text{vec}(Q_l^\top G Q_r)$. Thus, $F^{\text{soap}}(w)$ approximates $\hat{F}(w)$ via a structured, two-sided Kronecker factorization.

Alternatively, some optimizers construct preconditioners that directly approximate the Hessian rather than the whitening metric. For instance, AdaHessian [57] and Sophia [21] directly estimate quantities related to the diagonal of the Hessian using techniques such as Hutchinson's method and EMA. Their corresponding theoretical preconditioner takes the form $F(w) = |\operatorname{diag}(\nabla^2 f(w))|^p$, where $p > 0$ is a power parameter, and $|\cdot|$ denotes the element-wise absolute value.

Momentum-based Preconditioners. We take Muon as a representative example. At the k-th iteration, the momentum is updated as $M_k = \theta_{\text{muon}} M_{k-1} + G_k$, where $G_k \in \mathbb{R}^{m\times n}$ (assuming $m \geq n$). We have $F_k = (M_k^\top M_k)^{1/2} \otimes I_m$ (or replacing M_k with its Nesterov-accelerated version). Expanding this recurrence yields $M_k = \sum_{i=0}^{k} \theta_{\text{muon}}^{k-i} G_i$. Analyzing the second moment, we have:

$$\mathbb{E}[M_k^\top M_k] = \sum_{i,j\leq k} \theta_{\text{muon}}^{2k-(i+j)} \mathbb{E}[G_i^\top G_j] \approx \sum_{i=0}^{k} \theta_{\text{muon}}^{2(k-i)} \mathbb{E}[G_i^\top G_i].$$

Here, we assume that in the middle to late stages of training, the deterministic gradient component is negligible compared to the stochastic noise (or second moment magnitude), and that gradient noise is approximately independent across different steps i, j ($i \neq j$). Applying the same EMA perspective as used for AdamW and assuming the noise structure varies slowly with respect to w, $M_k^\top M_k$ serves as a stable estimator of $\frac{1}{1-\theta_{\text{muon}}^2}\mathbb{E}[G^\top G]$. Similarly, for the Nesterov-corrected momentum $\widetilde{M}_k = M_k + G_k/(1-\theta_{\text{muon}})$, we can similarly get that $\widetilde{M}_k^\top \widetilde{M}_k$ approximates $c\mathbb{E}[G^\top G]$ for some constant scaling factor c. Consequently, the implicit metric for Muon can be characterized as $F^{\text{muon}}(w) \propto (\mathbb{E}[G^\top G])^{\frac{1}{2}} \otimes I_m$, representing a single-sided Kronecker-factored approximation of $\hat{F}(w)$. This derivation extends naturally to element-wise operations on momentum; for instance, Lion's update rule implies a metric $F^{\text{lion}}(w) \propto \operatorname{diag}(\hat{F}(w))$.

J Proof

Notations A map is said to be smooth if it is infinitely differentiable. Given a matrix-valued function $F : \mathbb{R}^p \to \mathbb{R}^{p\times p}$, its directional derivative is written as $\nabla F(w)[v] = \lim_{\epsilon\to 0}(F(w+\epsilon v) - F(w))/\epsilon$. For notational brevity, we define $F_s(w) = P_s(w)F(w)P_s(w)$, $F_f(w) = P_f(w)F(w)P_f(w)$ and the corresponding pseudo inverse $F_s^\dagger(w) = P_s(w)F^{-1}(w)P_s(w)$, $F_f^\dagger(w) = P_f(w)F^{-1}(w)P_f(w)$. Denote $\kappa_{F_s} := \rho/\lambda_{F_s}$, $\kappa_F := \rho/\lambda_F$. The full version of Assumption 2.2 concerning the operator norm bounds on ∇P_s and ∇F is as follows. We further define $\delta = \sup_{w\in U} \|\nabla P_s(w)\|_{\text{op}}$ and $\delta_F = \sup_{w\in U} \|\nabla F(w)\|_{\text{op}}$. Then $\max\{\delta, \delta_F\} \leq \varepsilon$, and for any $u, v, w \in \mathbb{R}^p$, we have (according to Assumption 2)

$$\begin{aligned} \|\nabla P_s(w)[u]v\|_{F(w)} &\leq \frac{\delta}{G}\|u\|_{F(w)}\|v\|_{F(w)}, \\ \|\nabla F(w)[u]v\|_{F^{-1}(w)} &\leq \frac{\delta_F}{G}\|u\|_{F(w)}\|v\|_{F(w)}. \end{aligned} \tag{24}$$

J.1 Landscape Analysis

In this subsection we mainly focus on properties of Φ and $\mathcal{R}$.

Lemma 2. *Suppose that Assumption 2 holds. Define $\delta_{F_s} = 2\delta + \delta_F$. Then we have*

$$\max\{\|\nabla F_s^\dagger(w)[u]v\|_{F(w)}, \|\nabla F_f^\dagger(w)[u]v\|_{F(w)}\} \leq \frac{\delta_{F_s}}{G}\|u\|_{F(w)}\|v\|_{F^{-1}(w)},$$

$$\max\{\|\nabla F_s(w)[u]v\|_{F(w)}, \|\nabla F_f(w)[u]v\|_{F(w)}\} \leq \frac{\delta_{F_f}}{G}\|u\|_{F(w)}\|v\|_{F(w)}.$$

Proof. By (24), we have

$$\begin{aligned} v^\top \nabla(P_s F^{-1} P_s)(w)[u]v =& 2v^\top \nabla P_s(w)[u]F^{-1}(w)P_s(w)v + v^\top P_s(w)\nabla(F^{-1})(w)[u]P_s(w)v \\ \le& 2\|v\|_{F(w)^{-1}} \|\nabla P_s(w)[u]F(w)^{-1}P_s(w)v\|_{F(w)} \\ &+ \|P_s(w)F(w)^{-1}v\|_{F(w)} \|\nabla F(w)[u]F(w)^{-1}P_s(w)v\|_{F(w)^{-1}} \\ \le& \frac{\delta_{F_s}}{\|\nabla f(w)\|_{F(w)^{-1}}} \|v\|_{F(w)^{-1}} \|v\|_{F_s^\dagger(w)} \|u\|_{F(w)}, \end{aligned} \tag{25}$$

where the first inequality uses $\nabla(F^{-1})(w)[u] = -F^{-1}(w)\nabla F(w)[u]F^{-1}(w)$.

Similarly, for any p,

$$\begin{aligned} & p^\top \nabla(P_s F P_s)(w)[u]v \\ =& p^\top \nabla P_s(w)[u]F(w)P_s(w)v + p^\top P_s(w)F(w)\nabla P_s(w)[u]v + p^\top P_s(w)\nabla F(w)[u]P_s(w)v \\ \le& \|\nabla P_s(w)[u]v\|_{F(w)} \|F(w)P_s(w)p\|_{F(w)^{-1}} + \|\nabla P_s(w)[u]p\|_{F(w)} \|F(w)P_s(w)v\|_{F(w)^{-1}} \\ &+ \|P_s(w)p\|_{F(w)} \|\nabla F(w)[u]P_s(w)v\|_{F(w)^{-1}} \\ \le& \frac{\delta(\|v\|_{F_s(w)}\|p\|_{F(w)} + \|v\|_{F(w)}\|p\|_{F_s(w)}) + \delta_F \|v\|_{F_s(w)}\|p\|_{F_s(w)}}{\|\nabla f(w)\|_{F(w)^{-1}}} \|u\|_{F(w)}. \end{aligned} \tag{26}$$

□

Using $\nabla P_f = -\nabla P_s$, we can similarly get

$$v^\top \nabla(P_f F^{-1} P_f)(w)[u]v \le \frac{\delta_{F_s}}{\|\nabla f(w)\|_{F(w)^{-1}}} \|v\|_{F(w)^{-1}} \|v\|_{F_f^\dagger(w)} \|u\|_{F(w)},$$

and

$$v^\top \nabla(P_f F P_f)(w)[u]v \le \frac{\delta_{F_s}}{\|\nabla f(w)\|_{F(w)^{-1}}} \|v\|_{F(w)} \|v\|_{F_f(w)} \|u\|_{F(w)}. \tag{27}$$

Lemma 3. *Suppose Assumptions 1 and 2 hold. For any $w \in U$, we have*

$$\nabla\Phi(w)P_s(w)F^{-1}(w)\nabla f(w) = 0, \tag{28}$$

and $\operatorname{Range}\nabla\Phi(w) \subset T_{\Phi(w)}\mathcal{R}$. *Especially, if we further assume that* $\max\{\delta_F, \delta\} < \frac{1}{3}\lambda_{H_{F,s}}$, *then for any $z \in \mathcal{R}$, $\nabla\Phi(z)$ is the oblique projection onto $T_z\mathcal{R}$ in the direct sum decomposition* $\mathbb{R}^p = T_z\mathcal{R} \oplus \operatorname{Range} P_s(z)$.

Proof. First, note that $\Phi(w) = \lim_{n\to\infty} \phi_1^{(n)}(w)$, where $\phi_1(w) := \psi_t(w)|_{t=1}$ denotes the time-1 map of the flow. Since F and f are smooth, ϕ_1 is smooth. Then the smoothness of Φ follows immediately from Theorem 5.1 in [7]. For any $w \in U$, noting that

$$-\nabla\Phi(w)P_s(w)F^{-1}(w)\nabla f(w) = \frac{d}{dt}\Phi(\psi_t(w))|_{t=0} = \frac{d}{dt}\Phi(w)|_{t=0} = 0,$$

we get (28). Besides, $\operatorname{Range}\nabla\Phi(w) \subset T_{\Phi(w)}\mathcal{R}$ is due to $\Phi(w) \in \mathcal{R}$.

It follows that for any $z \in \mathcal{R}$ and any $v \in \mathbb{R}^p$, we have

$$\begin{aligned} 0 =& \nabla\left(\nabla\Phi F_s^\dagger \nabla f\right)(z)[v] \\ =& \nabla^2\Phi(z)[v]\underbrace{F_s^\dagger(z)\nabla f(z)}_{=0} + \nabla\Phi(z)\nabla F_s^\dagger(z)[v]\nabla f(z) + \nabla\Phi(z)F_s^\dagger(z)\nabla^2 f(z)[v] \\ =& \nabla\Phi(z)\left(\nabla F_s^\dagger(z)[v]\nabla f(z) + F_s^\dagger(z)\nabla^2 f(z)[v]\right). \end{aligned} \tag{29}$$

Define the linear map T_z as $T_z(v) = \nabla F_s^\dagger(z)[v]\nabla f(z) + F_s^\dagger(z)\nabla^2 f(z)v$. Noting that

$$\begin{aligned} \nabla F_s^\dagger(z)[v]\nabla f(z) =& \nabla P_s(z)[v]F^{-1}(z)P_s(z)\nabla f(z) + P_s(z)\nabla F^{-1}(z)[v]P_s(z)\nabla f(z) \\ &+ P_s(z)F^{-1}(z)\nabla P_s(z)[v]\nabla f(z) \\ =& P_s(z)F^{-1}(z)\nabla P_s(z)[v]\nabla f(z), \end{aligned}$$

the range of T_z is a subspace of $\operatorname{Range} P_s(z)$. By Lemma 2 and Assumption 2, for v satisfying $P_s(z)v = v$,

$$\|T_z(v)\|_{F(z)^{-1}} \geq \lambda_{H_{F,s}}\|v\|_{F(z)} - \frac{\delta_{F_s}}{G}\|\nabla f(z)\|_{F^{-1}(z)}\|v\|_{F(z)} \geq (\lambda_{H_{F,s}} - \delta_{F_s})\|v\|_{F(z)}.$$

This implies that $T_z|_{\operatorname{Range} P_s(z)} : \operatorname{Range} P_s(z) \to \operatorname{Range} P_s(z)$ is a linear isomorphism. Therefore, by (29), we get that $\operatorname{Range} P_s(z) \subset \operatorname{Null} \nabla\Phi(z)$, and $\dim(\operatorname{Null} \nabla\Phi(z)) \geq \dim(\operatorname{Range} P_s(z))$.

On the other hand, for any smoothed curve $\tilde{z}_t$ on $\mathcal{R}$, $\nabla\Phi(\tilde{z}_t)\dot{\tilde{z}}_t = \frac{d}{dt}\Phi(\tilde{z}_t) = \dot{\tilde{z}}_t$. Thus $\nabla\Phi(z)|_{T_z\mathcal{R}} = id$ for any $z \in \mathcal{R}$. It yields $p = \dim(\operatorname{Range} \nabla\Phi(z)) + \dim(\operatorname{Null} \nabla\Phi(z)) \geq \dim(T_z\mathcal{R}) + \dim(\operatorname{Range} P_s(z)) \geq p$, implying that $\operatorname{Range} P_s(z) = \operatorname{Null} \nabla\Phi(z)$. We get the conclusion. □

By definition, f exhibits strong convexity along the sharp directions, and consequently inherits certain properties of strongly convex functions. We first present this characteristic (as well as L-smoothness) in Lemmas 4 and 5.

Lemma 4. *Suppose Assumptions 1 and 2 hold. Denote $\gamma = \lambda_{H_F,s} - \frac{\delta_{F_s}}{2}$, $\ell = L_{H_F,s} + \frac{\delta_{F_s}}{2}$. Then for any $w \in U$ the following inequality hold*

$$\|P_s(\psi_t(w))F(\psi_t(w))^{-1}\nabla f(\psi_t(w))\|_{F(\psi_t(w))} \leq \exp(-\gamma t)\|P_s(w)F(w)^{-1}\nabla f(w)\|_{F(w)}, \quad (30)$$

and

$$\|P_s(\psi_t(w))F(\psi_t(w))^{-1}\nabla f(\psi_t(w))\|_{F(\psi_t(w))} \geq \exp(-\ell t)\|P_s(w)F(w)^{-1}\nabla f(w)\|_{F(w)}.$$

Proof.

$$\begin{aligned}\frac{d}{dt}\|P_sF^{-1}\nabla f\|_F^2 =& 2\partial_t\psi_t(w)^\top\nabla^2 f P_sF^{-1}\nabla f + \nabla f^\top \partial_t(P_sF^{-1}P_s)\nabla f\\ =& -2\nabla f^\top F^{-1}P_s\nabla^2 f P_sF^{-1}\nabla f + \nabla f^\top\nabla(F_s^\dagger)[-F_s^\dagger\nabla f]\nabla f\\ \overset{(25)}{\leq}& -2\lambda_{H_F,s}\|P_sF^{-1}\nabla f\|_F^2 + \delta_{F_s}\|P_sF^{-1}\nabla f\|_F^2.\end{aligned}$$

Using Gronwall's inequality (Lemma 15) gives

$$\|P_s(\psi_t(w))F(\psi_t(w))^{-1}\nabla f(\psi_t(w))\|_{F(\psi_t(w))}^2 \leq \exp(-2\gamma t)\|P_s(w)F(w)^{-1}\nabla f(w)\|_{F(w)}^2.$$

On the other hand, we have

$$\begin{aligned}\frac{d}{dt}\|P_sF^{-1}\nabla f\|_F^2 =& -2\nabla f^\top F^{-1}P_s\nabla^2 f P_sF^{-1}\nabla f + \nabla f^\top\nabla(F_s^\dagger)[-F_s^\dagger\nabla f]\nabla f\\ \overset{(25)}{\geq}& -2L_{H_F,s}\|P_sF^{-1}\nabla f\|_F^2 - \delta_{F_s}\|P_sF^{-1}\nabla f\|_F^2.\end{aligned}$$

Similarly, we get

$$\|P_s(\psi_t(w))F(\psi_t(w))^{-1}\nabla f(\psi_t(w))\|_{F(\psi_t(w))}^2 \geq \exp(-2\ell t)\|P_s(w)F(w)^{-1}\nabla f(w)\|_{F(w)}^2.$$

□

Lemma 5. *Suppose Assumptions 1 and 2 hold. Then for any $w \in U$ we have*

$$\frac{\gamma\lambda_{F_s}}{2}\|w - \Phi(w)\|_2^2 \leq f(w) - f(\Phi(w)), \quad (31)$$

and

$$\frac{1}{2\ell}\|P_s(w)F(w)^{-1}\nabla f(w)\|_{F(w)}^2 \leq f(w) - f(\Phi(w)) \leq \frac{1}{2\gamma}\|P_s(w)F(w)^{-1}\nabla f(w)\|_{F(w)}^2. \quad (32)$$

Proof. Given w, define $a_t = \|P_s(\psi_t(w))F(\psi_t(w))^{-1}\nabla f(\psi_t(w))\|_{F(\psi_t(w))}$. Then Lemma 4 implies that for any $t, s > 0$, $a_{t+s} \le e^{-\gamma s}a_t$. Note that $f(w) - f(\Phi(w)) = \int_0^\infty \nabla f(\psi_t(w))^\top P_s(\psi_t(w))F(\psi_t(w))^{-1}\nabla f(\psi_t(w))dt = \int_0^\infty a_t^2 dt$, $\|w - \Phi(w)\|_2 = \|\int_0^\infty P_s(\psi_t(w))F(\psi_t(w))^{-1}\nabla f(\psi_t(w))dt\|_2 \le \lambda_{F_s}^{-1/2}\int_0^\infty a_t dt$. We have

$$\begin{aligned}\lambda_{F_s}\|w-\Phi(w)\|_2^2 \le& \left(\int_0^\infty a_t dt\right)^2 = \int_0^\infty\int_0^\infty a_s a_t ds dt = 2\int_0^\infty dt\int_t^\infty a_s a_t ds\\ \le& 2\int_0^\infty a_t^2\left(\int_t^\infty e^{-\gamma(s-t)}ds\right)dt = \frac{2}{\gamma}\int_0^\infty a_t^2 dt.\end{aligned}$$

On the other hand, by Lemma 4, it holds that

$$\begin{aligned}f(w) - f(\Phi(w)) =& \int_0^\infty \|P_s(\psi_t(w))F(\psi_t(w))^{-1}\nabla f(\psi_t(w))\|^2_{F(\psi_t(w))}dt\\ \le& \int_0^\infty \exp(-2\gamma t)\|P_s(w)F(w)^{-1}\nabla f(w)\|^2_{F(w)}dt\\ =& \frac{1}{2\gamma}\|P_s(w)F(w)^{-1}\nabla f(w)\|^2_{F(w)},\end{aligned}$$

and

$$\begin{aligned}f(w) - f(\Phi(w)) =& \int_0^\infty \|P_s(\psi_t(w))F(\psi_t(w))^{-1}\nabla f(\psi_t(w))\|^2_{F(\psi_t(w))}dt\\ \ge& \int_0^\infty \exp(-2\ell t)\|P_s(w)F(w)^{-1}\nabla f(w)\|^2_{F(w)}dt\\ =& \frac{1}{2\ell}\|P_s(w)F(w)^{-1}\nabla f(w)\|^2_{F(w)}.\end{aligned}$$

□

The following lemma shows the property of $\nabla\Phi$: it approximates the projection to the flat subspace.

Lemma 6. *Suppose Assumptions 1 and 2 hold, and*

$$\mu_1 := \lambda_{H_F,s} - \frac{3}{2}\delta_{F_s} \ge 0, \quad 2\delta\lambda_{F_s}^{-1}\frac{1}{\gamma}\left(1+\frac{\delta_{F_s}}{2\mu_1}\right) \le \frac{1}{2\rho}.$$

Define

$$\epsilon_\Phi = \frac{4\delta\kappa_{F_s}}{\gamma+\mu_1} + \frac{4\kappa_{F_s}^2\delta}{\gamma}\left(1+\frac{\delta_{F_s}}{2\mu_1}\right)\left(1+\frac{2\delta}{\gamma+\mu_1}\right) + \frac{2\delta_{F_s}\kappa_{F_s}}{\mu_1}\left(1+\frac{2\delta}{\gamma+\mu_1}\right).$$

Then for any $w \in U$ and any vector $v \in \mathbb{R}^p$, we have

$$\|P_s(\Phi(w))\nabla\Phi(w)v\|_{F(\Phi(w))} \le \frac{\epsilon_\Phi}{2}\|v\|_{F(\Phi(w))},$$

$$\|P_f(\Phi(w))\nabla\Phi(w)v - P_f(w)v\|_{F(\Phi(w))} \le \frac{\epsilon_\Phi}{2}\|v\|_{F(\Phi(w))}.$$

Thus

$$\|\nabla\Phi(w)v - P_f(w)v\|_{F(\Phi(w))} \le \epsilon_\Phi\|v\|_{F(\Phi(w))}. \tag{33}$$

Proof. Define

$$h_t = P_s(\psi_t(w))\partial_w\psi_t(w)[v], \quad r_t = P_f(\psi_t(w))\partial_w\psi_t(w)[v].$$

It implies that $h_t + r_t = \partial_w\psi_t(w)[v]$, $h_0 + r_0 = v$ and $\lim_{t\to\infty} h_t + \lim_{t\to\infty} r_t = \nabla\Phi(w)[v]$.

We compute the time derivative of $\|h_t\|_F^2$:

$$\begin{aligned}
&\frac{d}{dt}\|h_t\|^2_{F(\psi_t(w))}\\
=&\frac{d}{dt}\left(v^\top \partial_w\psi_t(w)^\top F_s(\psi_t(w))\partial_w\psi_t(w)v\right)\\
=&\underbrace{v^\top \partial_w\psi_t(w)^\top \nabla F_s(\psi_t(w))[\partial_t\psi_t(w)]\partial_w\psi_t(w)v}_{A} + 2\underbrace{v^\top \partial_w\psi_t(w)^\top F_s(\psi_t(w))\partial_t\partial_w\psi_t(w)v}_{B}.
\end{aligned} \tag{34}$$

$$\begin{aligned}
A =& -v^\top \partial_w\psi_t(w)^\top \nabla F_s(\psi_t(w))[F_s^\dagger(\psi_t(w))\nabla f(\psi_t(w))]\partial_w\psi_t(w)v\\
\overset{(26)}{\leq}& \delta_{F_s}\|\partial_w\psi_t(w)v\|_{F_s}\|\partial_w\psi_t(w)v\|_F.
\end{aligned} \tag{35}$$

$$\begin{aligned}
B =& -h_t^\top F_s(\psi_t(w))\partial_w(F_s^\dagger(\psi_t(w))\nabla f(\psi_t(w)))v\\
=& -h_t^\top F_s(\psi_t(w))\partial_w(F_s^\dagger(\psi_t(w)))[v]\nabla f(\psi_t(w)) - h_t^\top \nabla^2 f(\psi_t(w))h_t\\
=& -h_t^\top F_s\nabla F_s^\dagger[\partial_w\psi_t(w)[v]]\nabla f - h_t^\top P_s\nabla^2 f P_s h_t\\
\overset{(25)}{\leq}& \delta_{F_s}\|F_s h_t\|_{F^{-1}}\|\partial_w\psi_t(w)[v]\|_F - \lambda_{H_F,s}\|h_t\|_F^2,
\end{aligned} \tag{36}$$

Substituting (35) and (36) into (34) yields

$$\begin{aligned}
\frac{d}{dt}\|h_t\|^2_{F(\psi_t(w))} &\leq 3\delta_{F_s}\|h_t\|_{F(\psi_t(w))}\|h_t + r_t\|_{F(\psi_t(w))} - 2\lambda_{H_F,s}\|h_t\|^2_{F(\psi_t(w))}\\
&\leq -2\underbrace{\left(\lambda_{H_F,s} - \frac{3}{2}\delta_{F_s}\right)}_{:=\mu_1}\|h_t\|^2_{F(\psi_t(w))} + \delta_{F_s}\|h_t\|_{F(\psi_t(w))}\|r_t\|_{F(\psi_t(w))}.
\end{aligned} \tag{37}$$

Next we consider the time derivative for r_t. Directly computing $\frac{d}{dt}r_t$ gives:

$$\frac{d}{dt}r_t = \underbrace{\nabla P_f(\psi_t(w))[\partial_t\psi_t(w)]\partial_w\psi_t(w)v}_{c} + \underbrace{P_f(\psi_t(w))\partial_t\partial_w\psi_t(w)v}_{d}. \tag{38}$$

For the term c:

$$\begin{aligned}
&\|c\|_{F(\psi_t(w))}\\
=&\|\nabla P_f(\psi_t(w))[F_s^\dagger(\psi_t(w))\nabla f(\psi_t(w))]\partial_w\psi_t(w)v\|_{F(\psi_t(w))}\\
\overset{(24)}{\leq}& \frac{\delta}{\|F_s^\dagger(\psi_0(w))\nabla f(\psi_0(w))\|_{F(\psi_0(w))}}\|\partial_w\psi_t(w)v\|_{F(\psi_t(w))}\|F_s^\dagger(\psi_t(w))\nabla f(\psi_t(w))\|_{F(\psi_t(w))}\\
\overset{(4)}{\leq}& \delta e^{-\gamma t}\|\partial_w\psi_t(w)v\|_{F(\psi_t(w))}.
\end{aligned} \tag{39}$$

where the first inequality also uses $\nabla P_s + \nabla P_f = \nabla I = 0$.

Note that for any vector p,

$$\begin{aligned}
P_f(w)\nabla(F_s^\dagger)(w)[p] &= P_f(w)\left[\nabla P_s(w)[p]F^{-1}P_s + P_s\nabla(F^{-1})(w)[p]P_s + P_sF^{-1}\nabla P_s[p]\right]\\
&= P_f(w)\nabla(P_s)(w)[p]F^{-1}(w)P_s(w).
\end{aligned} \tag{40}$$

Thus

$$
\begin{aligned}
&\|d\|_{F(\psi_t(w))} \\
=&\|P_f(\psi_t(w))\partial_w(F_s^\dagger(\psi_t(w))\nabla f(\psi_t(w)))v\|_{F(\psi_t(w))} \\
\leq&\|P_f(\psi_t(w))\partial_w(F_s^\dagger(\psi_t(w)))[v]\nabla f(\psi_t(w))\|_{F(\psi_t(w))} \\
&+\|P_f(\psi_t(w))F_s^\dagger(\psi_t(w))\nabla^2 f(\psi_t(w))\partial_w\psi_t(w)v\|_{F(\psi_t(w))} \\
\overset{(40)}{=}&\|P_f(\psi_t(w))\nabla P_s(\psi_t(w))[\partial_w\psi_t(w)v]F_s^\dagger(\psi_t(w))\nabla f(\psi_t(w))\|_{F(\psi_t(w))} \\
\overset{(24)}{\leq}&\frac{\delta}{\|F_s^\dagger(\psi_0(w))\nabla f(\psi_0(w))\|_{F(\psi_0(w))}}\|\partial_w\psi_t(w)[v]\|_{F(\psi_t(w))}\|F_s^\dagger(\psi_t(w))\nabla f(\psi_t(w))\|_{F(\psi_t(w))} \\
\overset{(30)}{\leq}&\delta e^{-\gamma t}\|\partial_w\psi_t(w)[v]\|_{F(\psi_t(w))}.
\end{aligned}
\tag{41}
$$

Substituting (39) and (41) into (38) yields

$$
\left\|\frac{d}{dt}r_t\right\|_{F(\psi_t(w))} \leq 2\delta e^{-\gamma t}\|h_t + r_t\|_{F(\psi_t(w))}. \tag{42}
$$

Now we combine (37) and (42) to get respective bounds. Denote $M_t = \sup_{0\leq s\leq t}\|r_s\|_{F(\psi_s(w))}$. Using Lemma 14 to (37) gives

$$
\begin{aligned}
\|h_t\|_{F(\psi_t(w))} &\leq e^{-\mu_1 t}\|h_0\|_{F(\psi_0(w))} + \frac{1}{2}\delta_{F_s}e^{-\mu_1 t}\int_0^t e^{\mu_1 s}\|r_s\|_{F(\psi_s(w))}ds \\
&\leq e^{-\mu_1 t}\|h_0\|_{F(\psi_0(w))} + \frac{\delta_{F_s}}{2\mu_1}M_t.
\end{aligned}
\tag{43}
$$

Thus for any $t \geq 0$,

$$
\begin{aligned}
\|r_t - r_0\|_2 \leq \int_0^t \left\|\frac{d}{dr}r_s\right\|_2 ds &\leq \lambda_{F_s}^{-1}\int_0^t \left\|\frac{d}{dr}r_s\right\|_{F(\psi_s(w))} ds \\
&\overset{(42)}{\leq}\lambda_{F_s}^{-1}\int_0^t 2\delta e^{-\gamma s}(\|h_s\|_{F(\psi_s(w))} + \|r_s\|_{F(\psi_s(w))})ds \\
&\overset{(43)}{\leq}2\delta\lambda_{F_s}^{-1}\int_0^t e^{-\gamma s}\left(e^{-\mu_1 s}\|h_0\|_{F(w)} + \left(1+\frac{\delta_{F_s}}{2\mu_1}\right)M_t\right)ds \\
&\leq 2\delta\lambda_{F_s}^{-1}\left(\frac{\|h_0\|_{F(w)}}{\gamma+\mu_1} + \frac{1}{\gamma}\left(1+\frac{\delta_{F_s}}{2\mu_1}\right)M_t\right).
\end{aligned}
\tag{44}
$$

Using $M_t \leq \rho(\|r_0\|_2 + \sup_{0\leq s\leq t}\|r_s - r_0\|_2)$ and $2\delta\lambda_{F_s}^{-1}\frac{1}{\gamma}\left(1+\frac{\delta_{F_s}}{2\mu_1}\right) \leq \frac{1}{2\rho}$, we get for any $t \geq 0$,

$$
M_t \leq 2\kappa_{F_s}\left(\|r_0\|_{F(w)} + 2\delta\frac{\|h_0\|_{F(w)}}{\gamma+\mu_1}\right).
$$

Substituting it into (43) and (44) respectively, we get

$$
\begin{aligned}
\|h_t\|_{F(\psi_t(w))} \leq& e^{-\mu_1 t}\|h_0\|_{F(\psi_0(w))} + \frac{\delta_{F_s}}{\mu_1}\kappa_{F_s}\left(\|r_0\|_{F(w)} + 2\delta\frac{\|h_0\|_{F(w)}}{\gamma+\mu_1}\right) \\
\leq& e^{-\mu_1 t}\|h_0\|_{F(\psi_0(w))} + \frac{1}{2}\epsilon_\Phi\|v\|_{F(w)},
\end{aligned}
$$

and

$$\begin{aligned}\|r_t - r_0\|_2 \leq & 2\delta\lambda_{F_s}^{-1}\frac{\|h_0\|_{F(w)}}{\gamma+\mu_1} + \frac{4\kappa_{F_s}^2\delta}{\gamma\rho}\left(1+\frac{\delta_{F_s}}{2\mu_1}\right)\left(\|r_0\|_{F(w)} + 2\delta\frac{\|h_0\|_{F(w)}}{\gamma+\mu_1}\right)\\ \leq & \frac{1}{2}\epsilon_\Phi\frac{\|v\|_{F(w)}}{\rho}.\end{aligned}$$

Noting that $\lim_{t\to\infty} h_t = P_s(\Phi(w))\nabla\Phi(w)v$ and $\lim_{t\to\infty}(r_t - r_0) = P_f(\Phi(w))\nabla\Phi(w)v - P_f(w)v$, we get the conclusion.

□

J.2 Dynamics Analysis

Lemma 7. *Suppose Assumptions 1 and 2 hold, and*

$$\epsilon_\Phi \leq \frac{1}{10}\min\{1, \alpha\beta_1\},\ \delta_{F_s} \leq \frac{1}{10}\lambda_{H_{F,s}},\ \eta_0\delta_{F_s}\chi\left(\frac{\kappa_F^{1/2}}{\alpha}+\beta_2\right) \leq \frac{1}{5}\alpha. \tag{45}$$

Let $\iota_t = \min\left\{\frac{1}{2}\alpha, \lambda_{H_F,s}\beta_1\eta_t\right\}$. *We have*

$$\begin{aligned}\|w_t - \Phi(w_t)\|_2^2 + \eta_t\|m_t\|_{F_s^\dagger(w_t)}^2 \lesssim & \exp\left(-\int_0^t \iota_s ds\right)\\ & + \varepsilon\int_0^t \exp\left(-\int_s^t \iota_\tau d\tau\right)(\|\nabla f(w_s)\|_{F_f^\dagger(w_s)}^2 + \|m_s\|_{F_f^\dagger(w_s)}^2)ds\end{aligned} \tag{46}$$

Especially, taking $\alpha = 2\eta_0\lambda_{H_{F,s}}^{\frac{1}{2}}$ *and* $\beta_1 = \lambda_{H_{F,s}}^{-\frac{1}{2}}$ *gives*

$$\begin{aligned}&\|w_t - \Phi(w_t)\|_2^2 + \eta_t\|m_t\|_{F_s^\dagger(w_t)}^2\\ \lesssim & \exp\left(-\lambda_{H_{F,s}}^{\frac{1}{2}}\int_0^t \eta_s ds\right) + \varepsilon\int_0^t \exp\left(-\lambda_{H_{F,s}}^{\frac{1}{2}}\int_s^t \eta_\tau d\tau\right)(\|\nabla f(w_s)\|_{F_f^\dagger(w_s)}^2 + \|m_s\|_{F_f^\dagger(w_s)}^2)ds.\end{aligned}$$

Remark 7. *Such choices of* α *and* β_1 *yield fast convergence in the (strongly convex) sharp directions of the ill-conditioned landscape. The rate* $\exp\left(-\lambda_{H_{F,s}}^{\frac{1}{2}}\int_0^t \eta_s ds\right)$ *matches the results in [3].*

Proof. We first consider general hyper-parameters α, β_1 and η_t. Recall that $\dot{w}_t = -\eta_t F_s(w_t)^{-1}(m_t + \beta_1\nabla f(w_t)) - \eta_t\chi F_f(w_t)^{-1}(m_t + \beta_2\nabla f(w_t))$. We begin by bounding $\|\dot{w}_t\|_{F(w_t)}$. Using $\dot{m}_t = -\alpha m_t + \nabla f(w_t)$, we have $m_t = e^{-\alpha t}\int_0^t e^{\alpha s}\nabla f(w_s)ds$. It yields

$$\begin{aligned}\|\dot{w}_t\|_{F(w_t)} \leq & \eta_t\left(\chi\|m_t\|_{F^{-1}(w_t)} + \beta_2\|\nabla f(w_t)\|_{F^{-1}(w_t)}\right)\\ \leq & \eta_t\chi\left(\lambda_F^{-\frac{1}{2}}\int_0^t e^{-\alpha(t-s)}\|\nabla f(w_s)\|_2 ds + \beta_2\|\nabla f(w_t)\|_{F^{-1}(w_t)}\right)\\ \leq & \underbrace{\eta_t\,\chi\left(\frac{\kappa_F^{1/2}}{\alpha}+\beta_2\right)}_{:=C_w} G.\end{aligned} \tag{47}$$

Now consider the Lyapunov function:

$$V_t = (f(w_t) - f(\Phi(w_t))) + \frac{\eta_t}{2}\|m_t\|_{F_s^\dagger(w_t)}^2.$$

For notational brevity, we define $\widetilde{m}_t = m_t + \beta_1\nabla f(w_t)$, and

$$\begin{aligned}\Delta_1 = & \langle\nabla f(w_t), (P_f(w_t) - \nabla\Phi(w_t))\dot{w}_t\rangle\,, \quad \Delta_2 = \frac{\dot{\eta}_t}{2}\|m_t\|_{F_s^\dagger(w_t)}^2,\\ \Delta_3 = & m_t^\top\nabla F_s^\dagger(w_t)[\dot{w}_t]m_t.\end{aligned}$$

For Δ_1 We have

$$\begin{aligned}\Delta_1 =& \langle \nabla f(w_t), (P_f(w_t) - \nabla\Phi(w_t))P_f(w_t)\dot{w}_t\rangle + \langle \nabla f(w_t), (P_f(w_t) - \nabla\Phi(w_t))P_s(w_t)\dot{w}_t\rangle \\ \overset{(33)}{\leq}& \epsilon_\Phi\eta_t\|\nabla f(w_t)\|_{F^{-1}(w_t)}\left(\|m_t + \beta_1\nabla f(w_t)\|_{F_s^\dagger(w_t)} + \chi\|m_t + \beta_2\nabla f(w_t)\|_{F_f^\dagger(w_t)}\right) \\ \leq& \epsilon_\Phi\eta_t(\|\nabla f(w_t)\|_{F_s^\dagger(w_t)} + \|\nabla f(w_t)\|_{F_f^\dagger(w_t)})\left(\|m_t\|_{F_s^\dagger(w_t)} + \beta_1\|\nabla f(w_t)\|_{F_s^\dagger(w_t)}\right) \\ &+ \epsilon_\Phi\eta_t(\|\nabla f(w_t)\|_{F_s^\dagger(w_t)} + \|\nabla f(w_t)\|_{F_f^\dagger(w_t)})\left(\chi\|m_t\|_{F_f^\dagger(w_t)} + \chi\beta_2\|\nabla f(w_t)\|_{F_f^\dagger(w_t)}\right) \\ \leq& 2\epsilon_\Phi(\frac{1}{\alpha} + \beta_1)\eta_t\|\nabla f(w_t)\|^2_{F_s^\dagger(w_t)} + \epsilon_\Phi(\frac{1}{\alpha} + \frac{\beta_1}{2} + \chi\beta_2 + \frac{\chi^2\beta_2^2\alpha}{2})\eta_t\|\nabla f(w_t)\|^2_{F_f^\dagger(w_t)} \\ &+ \frac{\epsilon_\Phi\alpha}{2}\eta_t\|m_t\|^2_{F_s^\dagger(w_t)} + \frac{\epsilon_\Phi\alpha}{2}\eta_t\chi^2\|m_t\|^2_{F_f^\dagger(w_t)}.\end{aligned} \tag{48}$$

where the last inequality uses AM-GM inequality.

For Δ_2,

$$\begin{aligned}|\Delta_3| \overset{(25)}{\leq}& \eta_t\delta_{F_s}C_w\|m_t\|_{F_s^\dagger(w_t)}\|m_t\|_{F^{-1}(w_t)} \\ \leq& \eta_t\delta_{F_s}C_w\|m_t\|^2_{F_s^\dagger(w_t)} + \eta_t\delta_{F_s}C_w\|m_t\|_{F_s^\dagger(w_t)}\|m_t\|_{F_f^\dagger(w_t)} \\ \leq& \frac{3}{2}\eta_t\delta_{F_s}C_w\|m_t\|^2_{F_s^\dagger(w_t)} + \frac{1}{2}\eta_t\delta_{F_s}C_w\|m_t\|^2_{F_f^\dagger(w_t)},\end{aligned} \tag{49}$$

where the second inequality uses $\|m_t\|_{F^{-1}(w_t)} \leq \|m_t\|_{F_s^\dagger(w_t)} + \|m_t\|_{F_f^\dagger(w_t)}$ and the last inequality uses Cauchy inequality.

Directly computing the time derivative yields

$$\begin{aligned}\frac{d}{dt}V_t + \iota_t V_t =& \Delta_1 + \Delta_2 + \frac{\eta_t}{2}\Delta_3 - \alpha\eta_t\|m_t\|^2_{F_s^\dagger(w_t)} - \beta_1\eta_t\|\nabla f(w_t)\|^2_{F_s^\dagger(w_t)} \\ &+ \iota_t\left(f(w_t) - f(\Phi(w_t))\right) + \frac{\eta_t\iota_t}{2}\|m_t\|^2_{F_s^\dagger(w_t)} \\ \leq& \left(-\beta_1\eta_t + \frac{\iota_t}{2\gamma} + 2\epsilon_\Phi(\frac{1}{\alpha} + \beta_1)\eta_t\right)\|\nabla f(w_t)\|^2_{F_s^\dagger(w_t)} \\ &+ \left(-\alpha\eta_t + \frac{\eta_t\iota_t}{2} + \frac{1}{2}\epsilon_\Phi\alpha\eta_t + \eta_t^2\delta_{F_s}C_w\right)\|m_t\|^2_{F_s^\dagger(w_t)} \\ &+ \epsilon_\Phi(\frac{1}{\alpha} + \frac{\beta_1}{2} + \chi\beta_2 + \frac{\chi^2\beta_2^2\alpha}{2})\eta_t\|\nabla f(w_t)\|^2_{F_f^\dagger(w_t)} \\ &+ (\frac{1}{4}\eta_t^2\delta_{F_s}C_w + \frac{\epsilon_\Phi\alpha}{2}\eta_t\chi^2)\|m_t\|^2_{F_f^\dagger(w_t)},\end{aligned}$$

where the inequality uses $\dot{\eta}_t \leq 0$, (32), (48) and (49).

Now set $\epsilon_u = \epsilon_\Phi(\frac{1}{\alpha} + \frac{\beta_1}{2} + \chi\beta_2 + \frac{\chi^2\beta_2^2\alpha}{2})\eta_t + (\frac{1}{4}\eta_t^2\delta_{F_s}C_w + \frac{\epsilon_\Phi\alpha}{2}\eta_t\chi^2)$ and $U_t = \|\nabla f(w_t)\|^2_{F_f^\dagger(w_t)} + \|m_t\|^2_{F_f^\dagger(w_t)}$.

Then by conditions in (45), we have

$$\frac{d}{dt}V_t + \iota_t V_t \leq \epsilon_u U_t.$$

Integrating $\frac{d}{dt}(e^{\int_0^t \iota_s ds}V_t)$ yields

$$e^{\int_0^t \iota_s ds}V_t \leq V_0 + \epsilon_u\int_0^t e^{\int_0^s \iota_\tau d\tau}U_s ds.$$

By (31) in Lemma 5, we obtain

$$
\begin{aligned}
&\|w_t - \Phi(w_t)\|_2^2 + \eta_t \|m_t\|^2_{F_s^\dagger(w_t)} \\
\lesssim &\exp\left(-\int_0^t \iota_s ds\right) + \varepsilon \int_0^t \exp\left(-\int_s^t \iota_\tau d\tau\right) (\|\nabla f(w_s)\|^2_{F_f^\dagger(w_s)} + \|m_s\|^2_{F_f^\dagger(w_s)}) ds.
\end{aligned}
$$

If we further take $\alpha = 2\eta_0 \lambda^{\frac{1}{2}}_{H_{F,s}}$ and $\beta_1 = \lambda^{-\frac{1}{2}}_{H_{F,s}}$, then $\iota_t = \eta_t \lambda^{\frac{1}{2}}_{H_{F,s}}$, and we finally get

$$
\begin{aligned}
&\|w_t - \Phi(w_t)\|_2^2 + \eta_t \|m_t\|^2_{F_s^\dagger(w_t)} \\
\lesssim &\exp\left(-\lambda^{\frac{1}{2}}_{H_{F,s}} \int_0^t \eta_s ds\right) + \varepsilon \int_0^t \exp\left(-\lambda^{\frac{1}{2}}_{H_{F,s}} \int_s^t \eta_\tau d\tau\right) (\|\nabla f(w_s)\|^2_{F_f^\dagger(w_s)} + \|m_s\|^2_{F_f^\dagger(w_s)}) ds.
\end{aligned}
$$

□

Lemma 8. *Suppose Assumptions 1 and 2 hold. For any $z \in \mathcal{R}$, we have*

$$
\|P_{\mathcal{R}}(z)v - P_f(z)v\|_{F(z)} \le \frac{\delta_{F_s}}{\lambda_{H_F,s}} \|v\|_{F(z)}.
$$

Proof. We first compute the closed form of $P_{\mathcal{R}}$. Let $\mathbf{0}^p$ denote the p dimensional all zero vector. Note that the $p-k$ dimensional manifold $\mathcal{R}$ is defined by $\mathcal{R} = \{w : P_s(w)\nabla f(w) = \mathbf{0}^p\} = \{w : F_s^\dagger(w)\nabla f(w) = \mathbf{0}^p\}$. Define $N(w) = F_s^\dagger(w)\nabla f(w)$. Thus the tangent space is $T_z\mathcal{R} = \{v : \nabla N(z)v = \mathbf{0}^p\}$. The projection to $T_z M$ is given by

$$
P_{\mathcal{R}}(z)v = \mathrm{argmin}_{u \in \mathbb{R}^p} \|v - u\|^2_{F(z)}, \quad \text{s.t. } \nabla N(z)u = \mathbf{0}^p.
$$

Directly computing ∇N gives $\nabla N(z)u = \nabla F_s^\dagger(w)[u]\nabla f(w) + F_s^\dagger(w)\nabla f^2(w)u$.

Noting that $\|v-u\|^2_{F(z)} = \|F(z)^{\frac{1}{2}}v - F(z)^{\frac{1}{2}}u\|_2^2$ and $\nabla N(z)u = \nabla N(z)F(z)^{-\frac{1}{2}}F(z)^{\frac{1}{2}}u$, We have

$$
F(z)^{\frac{1}{2}} P_{\mathcal{R}}(z)v = \underbrace{(I - F(z)^{-\frac{1}{2}}\nabla N(z)^\top (\nabla N(z)F(z)^{-1}\nabla N(z)^\top)^\dagger \nabla N(z) F(z)^{-\frac{1}{2}})}_{K(z)} F(z)^{\frac{1}{2}} v. \tag{50}
$$

For notational brevity, we define $J = F(z)^{-\frac{1}{2}} P_s(z) \nabla^2 f(z) P_s(z) F(z)^{-\frac{1}{2}}$, and $\Delta J \in \mathbb{R}^{k \times p}$ satisfying $\Delta J q = F(z)^{\frac{1}{2}} \nabla F_s^\dagger(w)[F(z)^{-\frac{1}{2}}q]^\top \nabla f(z)$ for any $q \in \mathbb{R}^p$.

Note that $K(z)$ is the Euclidean projection to the kernel subspace:

$$
\ker\left(\nabla N(z) F(z)^{-\frac{1}{2}}\right) = \ker\left(F(z)^{\frac{1}{2}} \nabla N(z) F(z)^{-\frac{1}{2}}\right) = \ker\left(J + \Delta J\right),
$$

and $P_f(z)$ is the Euclidean projection to $\ker J$. Using Wedin $\sin(\Theta)$ Theorem (Lemma 16) to J and $J + \Delta J$, we get

$$
\|K(z) - P_f(z)\|_2 \le \frac{\|\Delta J\|_2}{\lambda_{H_F,s}}.
$$

It follows that

$$
\begin{aligned}
\|F(z)^{\frac{1}{2}} P_{\mathcal{R}}(z)v - F(z)^{\frac{1}{2}} P_f(z)v\|_2 =& \|K(z)F(z)^{\frac{1}{2}}v - P_f(z)F(z)^{\frac{1}{2}}v\|_2 \\
\le& \frac{\|\Delta J\|_2}{\lambda_{H_F,s}} \|F(z)^{\frac{1}{2}} v\|_2.
\end{aligned} \tag{51}
$$

Note that

$$
\|\Delta J q\|_2 = \|\nabla F_s^\dagger(w)[F(z)^{-\frac{1}{2}}q]^\top \nabla f(z)\|_{F(z)} \overset{(25)}{\le} \delta_{F_s} \|q\|_2.
$$

Substituting it into (51) yields

$$
\|P_{\mathcal{R}}(z)v - P_f(z)v\|_{F(z)} = \|F(z)^{\frac{1}{2}} P_{\mathcal{R}}(z)v - F(z)^{\frac{1}{2}} P_f(z)v\|_2 \le \frac{\delta_{F_s}}{\lambda_{H_F,s}} \|v\|_{F(z)}.
$$

□

Lemma 9. *Suppose Assumptions 1 and 2 hold. For any $x, y \in U$ and $v \in \mathbb{R}^p$, we have*

$$\|F_s^\dagger(x)v - F_s^\dagger(y)v\|_2 \le \frac{\delta_{F_s}}{G}\kappa_F^{\frac{1}{2}}\lambda_F^{-\frac{1}{2}}\|x-y\|_2\|v\|_2, \tag{52}$$

$$\|F_f^\dagger(x)v - F_f^\dagger(y)v\|_2 \le \frac{\delta_{F_s}}{G}\kappa_F^{\frac{1}{2}}\lambda_F^{-\frac{1}{2}}\|x-y\|_2\|v\|_2, \tag{53}$$

$$\|P_f(x)v - P_f(y)v\|_2 \le \frac{\delta}{G}\kappa_F^{\frac{1}{2}}\lambda_F^{-\frac{1}{2}}\|x-y\|_2\|v\|_2. \tag{54}$$

Proof. By Assumption 2,

$$\|F_s^\dagger(x)v - F_s^\dagger(y)v\|_2 = \|\int_0^1 \nabla F_s^\dagger(x+(y-x)t)[x-y]vdt\|_2 \overset{(25)}{\le} \frac{\delta_{F_s}}{G}\kappa_F^{\frac{1}{2}}\lambda_F^{-\frac{1}{2}}\|x-y\|_2\|v\|_2$$

We can similarly prove (53) by (27). On the other hand, using $\nabla P_f + \nabla P_s = 0$, we get

$$\|P_f(x)v - P_f(y)v\|_2 = \|\int_0^1 \nabla P_s(x+(y-x)t)[x-y]vdt\|_2 \le \frac{\delta}{G}\kappa_F^{\frac{1}{2}}\lambda_F^{-\frac{1}{2}}\|x-y\|_2\|v\|_2.$$

□

Lemma 10. *Suppose Assumptions 1 and 2 hold. The dynamics of $z_t = \Phi(w_t)$ satisfies*

$$\begin{cases} \dot{z}_t = -\eta_t\chi P_{\mathcal{R}}(z_t)F(z_t)^{-1}(s_t + \beta_2\nabla f(z_t)) + \eta_t\epsilon_{z,t}, \\ \dot{s}_t = -\alpha s_t + \nabla f(z_t), \end{cases}$$

where $s_0 = 0$ and

$$\|\epsilon_{z,t}\|_2 \lesssim \int_0^t e^{-\alpha(t-s)}\|w_s - \Phi(w_s)\|_2 ds + \|w_t - \Phi(w_t)\|_2 + \varepsilon\|m_t\|_{F_s^\dagger(w_t)} + \varepsilon\|F_f^\dagger(z_t)(s_t + \beta_2\nabla f(z_t))\|_2 \tag{55}$$

Proof. By (28), the dynamics of $z_t = \Phi(w_t)$ satisfies

$$\dot{z}_t = -\eta_t\nabla\Phi(w_t)\left(F_s^\dagger(w_t)m_t + \chi F_f^\dagger(w_t)(m_t + \beta_2\nabla f(w_t))\right). \tag{56}$$

Define s_t satisfying $\dot{s}_t = -\alpha s_t + \nabla f(z_t)$ and $s_0 = 0$. Then

$$\|s_t - m_t\|_2 = \left\|\int_0^t e^{-\alpha(t-s)}(\nabla f(z_s) - \nabla f(w_s))ds\right\|_2 \le L\int_0^t e^{-\alpha(t-s)}\|w_s - \Phi(w_s)\|_2 ds \tag{57}$$

and

$$\begin{aligned} \max\{\|s_t\|_2, \|m_t\|_2\} &\le \int_0^t e^{-\alpha(t-s)}\max\{\|\nabla f(w_s)\|_2, \|\nabla f(z_s)\|_2\}ds \\ &\le \int_0^t e^{-\alpha(t-s)}\lambda_F^{-\frac{1}{2}}Gds \le \frac{\lambda_F^{-\frac{1}{2}}}{\alpha}G. \end{aligned} \tag{58}$$

On the other hand,

$$\begin{aligned} \|F_s^\dagger(w_t)m_t - F_s^\dagger(z_t)s_t\|_2 \le& \|F_s^\dagger(w_t)(m_t - s_t)\|_2 + \|F_s^\dagger(w_t)s_t - F_s^\dagger(z_t)s_t\|_2 \\ \overset{(52)}{\le}& \lambda_{F_s}^{-1}\|m_t - s_t\|_2 + \frac{\delta_{F_s}}{G}\kappa_F^{\frac{1}{2}}\lambda_F^{-\frac{1}{2}}\|s_t\|_2\|w_t - \Phi(w_t)\|_2 \\ \overset{(58)}{\le}& \lambda_{F_s}^{-1}\|m_t - s_t\|_2 + \frac{\delta_{F_s}}{\alpha}\kappa_F^{\frac{1}{2}}\lambda_F^{-1}\|w_t - \Phi(w_t)\|_2 \end{aligned} \tag{59}$$

Similarly

$$\begin{aligned}
&\|F_f^\dagger(w_t)m_t - F_f^\dagger(z_t)s_t\|_2 \\
\leq&\|F_f^\dagger(w_t)(m_t - s_t)\|_2 + \|F_f^\dagger(w_t)s_t - F_f^\dagger(z_t)s_t\|_2 \\
\overset{(53)}{\leq}&\lambda_F^{-1}\|m_t - s_t\|_2 + \frac{\delta_{F_s}}{G}\kappa_F^{\frac{1}{2}}\lambda_F^{-\frac{1}{2}}\|s_t\|_2\|w_t - \Phi(w_t)\|_2 \\
\overset{(58)}{\leq}&\lambda_{F_s}^{-1}\|m_t - s_t\|_2 + \frac{\delta_{F_s}}{\alpha}\kappa_F^{\frac{1}{2}}\lambda_F^{-1}\|w_t - \Phi(w_t)\|_2,
\end{aligned} \tag{60}$$

and

$$\begin{aligned}
&\|F_f^\dagger(w_t)\nabla f(w_t) - F_f^\dagger(z_t)\nabla f(z_t)\|_2 \\
\leq&\|F_f^\dagger(w_t)(\nabla f(w_t) - \nabla f(z_t))\|_2 + \|F_f^\dagger(w_t)\nabla f(z_t) - F_f^\dagger(z_t)\nabla f(z_t)\|_2 \\
\overset{(53)}{\leq}&\lambda_F^{-1}L\|w_t - \Phi(w_t)\|_2 + \delta_{F_s}\kappa_F^{\frac{1}{2}}\lambda_F^{-1}\|w_t - \Phi(w_t)\|_2.
\end{aligned} \tag{61}$$

Now we rewrite (56) as

$$\dot{z}_t = -\eta_t P_{\mathcal{R}}(z_t)\left(F_s^\dagger(z_t)s_t + \chi F_f^\dagger(z_t)(s_t + \beta_2\nabla f(z_t))\right) + \eta_t\epsilon_{z,t},$$

where

$$\begin{aligned}
\epsilon_{z,t} = &-\underbrace{P_{\mathcal{R}}(z_t)\left(F_s^\dagger(w_t)m_t - F_s^\dagger(z_t)s_t\right)}_{a} \\
&-\underbrace{(\nabla\Phi(w_t) - P_{\mathcal{R}}(z_t))\left(F_s^\dagger(w_t)m_t + \chi F_f^\dagger(z_t)(s_t + \beta_2\nabla f(z_t))\right)}_{b} \\
&-\underbrace{\chi\nabla\Phi(w_t)\left(F_f^\dagger(w_t)(m_t + \beta_2\nabla f(w_t)) - F_f^\dagger(z_t)(s_t + \beta_2\nabla f(z_t))\right)}_{c}.
\end{aligned}$$

By (50), we have

$$\begin{aligned}
\|a\|_2 \leq&\lambda_F^{-\frac{1}{2}}\|a\|_{F(z_t)} \leq \lambda_F^{-\frac{1}{2}}\|F_s^\dagger(w_t)m_t - F_s^\dagger(z_t)s_t\|_{F(z_t)} \leq \kappa_F^{\frac{1}{2}}\|F_s^\dagger(w_t)m_t - F_s^\dagger(z_t)s_t\|_2 \\
\overset{(59)}{\leq}&\kappa_F^{\frac{1}{2}}\lambda_{F_s}^{-1}\|m_t - s_t\|_2 + \kappa_F^{\frac{1}{2}}\frac{\delta_{F_s}}{\alpha}\kappa_F^{\frac{1}{2}}\lambda_F^{-1}\|w_t - \Phi(w_t)\|_2.
\end{aligned} \tag{62}$$

For any v,

$$\begin{aligned}
&\|\nabla\Phi(w_t)v - P_{\mathcal{R}}(z_t)v\|_2 \\
\leq&\|\nabla\Phi(w_t)v - P_f(w_t)v\|_2 + \|P_f(w_t)v - P_f(z_t)v\|_2 + \|P_f(z_t)v - P_{\mathcal{R}}(z_t)v\|_2 \\
\leq&\lambda_F^{-\frac{1}{2}}\|\nabla\Phi(w_t)v - P_f(w_t)v\|_{F(w_t)} + \|P_f(w_t)v - P_f(z_t)v\|_2 + \lambda_F^{-\frac{1}{2}}\|P_f(z_t)v - P_{\mathcal{R}}(z_t)v\|_{F(w_t)} \\
\leq&\rho^{\frac{1}{2}}\lambda_F^{-\frac{1}{2}}(\epsilon_\Phi + \frac{\delta_{F_s}}{\lambda_{H_{F,s}}})\|v\|_2 + \frac{\delta}{G}\kappa_F^{\frac{1}{2}}\lambda_F^{-\frac{1}{2}}\|w_t - \Phi(w_t)\|_2\|v\|_2,
\end{aligned}$$

where the last inequality uses (54), Lemma (8) and (33). It follows that

$$\begin{aligned}
\|b\|_2 \leq&\kappa_F^{\frac{1}{2}}(\epsilon_\Phi + \frac{\delta_{F_s}}{\lambda_{H_{F,s}}})\|F_s^\dagger(w_t)m_t + \chi F_f^\dagger(z_t)(s_t + \beta_2\nabla f(z_t))\|_2 \\
&+ \frac{\delta}{G}\kappa_F^{\frac{1}{2}}\lambda_F^{-\frac{1}{2}}\|w_t - \Phi(w_t)\|_2\|F_s^\dagger(w_t)m_t + \chi F_f^\dagger(z_t)(s_t + \beta_2\nabla f(z_t))\|_2 \\
\overset{(58)}{\leq}&\kappa_F^{\frac{1}{2}}(\epsilon_\Phi + \frac{\delta_{F_s}}{\lambda_{H_{F,s}}})\|F_s^\dagger(w_t)m_t\|_2 + \chi\kappa_F^{\frac{1}{2}}(\epsilon_\Phi + \frac{\delta_{F_s}}{\lambda_{H_{F,s}}})\|F_f^\dagger(z_t)(s_t + \beta_2\nabla f(z_t))\|_2 \\
&+ 2\chi\delta\kappa_F^{\frac{1}{2}}\lambda_F^{-1}(\frac{1}{\alpha} + \beta_2)\|w_t - \Phi(w_t)\|_2.
\end{aligned} \tag{63}$$

By (33) we have $\|\nabla\Phi(w_t)\|_2 \leq \|\nabla\Phi(w_t) - P_f(w_t)\|_2 + \|P_f(w_t)\|_2 \leq 1 + \epsilon_\Phi\kappa_F$. Thus

$$
\begin{aligned}
\|c\|_2 \leq & \chi(1+\epsilon_\Phi\kappa_F)\|F_f^\dagger(w_t)(m_t+\beta_2\nabla f(w_t)) - F_f^\dagger(z_t)(s_t+\beta_2\nabla f(z_t))\|_2 \\
\overset{(60),(61)}{\leq} & \chi(1+\epsilon_\Phi\kappa_F)\lambda_{F_s}^{-1}\|m_t - s_t\|_2 \\
& + \chi(1+\epsilon_\Phi\kappa_F)(\frac{\delta_{F_s}}{\alpha}\kappa_F^{\frac{1}{2}}\lambda_F^{-1} + \beta_2\lambda_F^{-1}L + \beta_2\delta_{F_s}\kappa_F^{\frac{1}{2}}\lambda_F^{-1})\|w_t - \Phi(w_t)\|_2.
\end{aligned} \tag{64}
$$

Combining (62), (63) and (64), we get

$$
\|a+b+c\|_2 = \mathcal{O}(\|m_t - s_t\|_2 + \|w_t - \Phi(w_t)\|_2 + \varepsilon\|m_t\|_{F_s^\dagger(w_t)} + \varepsilon\|F_f^\dagger(z_t)(s_t+\beta_2\nabla f(z_t))\|_2)
$$

Then using (57) gives the conclusion. □

Lemma 11. *Suppose Assumptions 1 and 2 hold. Then there exist a constant $\varepsilon_0 > 0$ depending on f and $\alpha, \beta, \{\eta_t\}_t, \chi, \beta_2$ such that for any $\varepsilon < \varepsilon_0$, we have*

$$
\|\epsilon_{z,t}\|_2^2 \lesssim e^{-\Pi_t} + \varepsilon\int_0^t e^{\Pi_\tau - \Pi_t}\left(\|\nabla f(z_\tau)\|_{F_f^\dagger(z_t)}^2 + \|s_\tau\|_{F_f^\dagger(z_t)}^2\right)d\tau + \varepsilon^2\|s_t+\beta_2\nabla f(z_t)\|_{F_f^\dagger(z_t)}^2.
$$

Proof. Define $\Pi_s = \int_0^s \iota_\tau d\tau$, $D_t = \|w_t - \Phi(w_t)\|_2 + \eta_t^{1/2}\|m_t\|_{F_s^\dagger(w_t)}$ and $E_t = \|\nabla f(z_t)\|_{F_f^\dagger(z_t)}^2 + \|s_t\|_{F_f^\dagger(z_t)}^2$ for notational brevity. By (46) we have

$$
\begin{aligned}
D_t^2 \lesssim & \exp(-\Pi_t) + \varepsilon\int_0^t \exp(\Pi_s - \Pi_t)\left(\|\nabla f(w_s) - \nabla f(z_s)\|_{F_f^\dagger(z_t)}^2 + \|m_s - z_s\|_{F_f^\dagger(z_t)}^2 + E_s\right)ds \\
\overset{(57)}{\lesssim} & \exp(-\Pi_t) + \varepsilon\int_0^t \exp(\Pi_s - \Pi_t)\left(D_s^2 + \int_0^s e^{-\alpha(s-\tau)}D_\tau^2 d\tau + E_s\right)ds \\
\leq & \exp(-\Pi_t) + \varepsilon\int_0^t \exp(\Pi_s - \Pi_t)\left(D_s^2 + E_s\right)ds,
\end{aligned}
$$

where we also use $(\int_0^s e^{-\alpha(s-\tau)}D_\tau d\tau)^2 \leq \int_0^s e^{-\alpha(s-\tau)}D_\tau^2 d\tau \int_0^s e^{-\alpha(s-\tau)}d\tau \lesssim \int_0^s e^{-\alpha(s-\tau)}D_\tau^2 d\tau$ in the second inequality, and

$$
\begin{aligned}
\int_0^t \exp(\Pi_s - \Pi_t)\int_0^s e^{-\alpha(s-\tau)}D_\tau^2 d\tau ds &= \int_0^t \exp(\Pi_\tau - \Pi_t)D_\tau^2 d\tau \int_\tau^t e^{\alpha\tau - \alpha s}\exp(\underbrace{\Pi_s - \Pi_\tau}_{\leq (s-\tau)\iota_\tau})ds \\
&\leq \frac{1}{\alpha - \eta_t}\int_0^t \exp(\Pi_\tau - \Pi_t)D_\tau^2 d\tau.
\end{aligned}
$$

in the third inequality. Then by using Gronwall's inequality in Lemma 15 to $e^{\Pi_t}D_t^2$, we get

$$
\exp(\Pi_t)D_t^2 \lesssim \exp(\mathcal{O}(\varepsilon t))\left(1 + \varepsilon\int_0^t \exp(\Pi_s)E_s ds\right). \tag{65}
$$

Substituting it into (57) yields

$$
\begin{aligned}
& \left(\int_0^t e^{-\alpha(t-s)}\|w_s - \Phi(w_s)\|_2 ds\right)^2 \\
\leq & \int_0^t e^{-\alpha(t-s)}ds\int_0^t e^{-\alpha(t-s)}\|w_s - \Phi(w_s)\|_2^2 ds \\
\lesssim & \int_0^t e^{-\alpha(t-s)}\exp(-\Pi_s)ds + \varepsilon\int_0^t e^{-\alpha(t-s)}ds\int_0^s \exp(\Pi_\tau - \Pi_s)E_\tau d\tau.
\end{aligned} \tag{66}
$$

To bound the right hand side, we first note that for any fixed $0 \leq \tau < t$, $\frac{\alpha}{2} \geq \frac{\Pi_s - \Pi_\tau}{s-\tau} \geq \frac{\Pi_t - \Pi_\tau}{t-\tau} := \bar{\iota}$ since ι_t is non-increasing. This gives

$$
\int_\tau^t e^{-\alpha(t-s)}\exp(\Pi_\tau - \Pi_s)ds \leq \int_\tau^t e^{-\alpha(t-s)}e^{-\bar{\iota}(s-\tau)}ds \leq \frac{1}{\alpha - \bar{\iota}}e^{-\bar{\iota}(t-\tau)} \leq \frac{2}{\alpha}e^{\Pi_\tau - \Pi_t}. \tag{67}
$$

Thus for the first term in (66), we have

$$\int_0^t e^{-\alpha(t-s)} \exp\left(-\Pi_s\right) ds \leq \frac{2}{\alpha} e^{-\Pi_t}. \tag{68}$$

For the second term in (66),

$$\begin{aligned}\int_0^t e^{-\alpha(t-s)} ds \int_0^s \exp\left(\Pi_\tau - \Pi_s\right) E_\tau d\tau &= \int_0^t E_\tau d\tau \int_\tau^t e^{-\alpha(t-s)} \exp\left(\Pi_\tau - \Pi_s\right) ds \\ &\overset{(67)}{\leq} \frac{2}{\alpha} \int_0^t \exp\left(\Pi_\tau - \Pi_t\right) E_\tau d\tau.\end{aligned} \tag{69}$$

Substituting (68) and (69) into (66), we get

$$\left(\int_0^t e^{-\alpha(t-s)} \|w_s - \Phi(w_s)\|_2 ds\right)^2 \lesssim e^{-\Pi_t} + \varepsilon \int_0^t \exp\left(\Pi_\tau - \Pi_t\right) E_\tau d\tau.$$

Substituting it and (65) into (55) gives

$$\|\epsilon_{z,t}\|_2^2 \lesssim e^{-\Pi_t} + \varepsilon \int_0^t e^{\Pi_\tau - \Pi_t} \left(\|\nabla f(z_\tau)\|_{F_f^\dagger(z_t)}^2 + \|s_\tau\|_{F_f^\dagger(z_t)}^2\right) d\tau + \varepsilon^2 \|s_t + \beta_2 \nabla f(z_t)\|_{F_f^\dagger(z_t)}^2.$$

Note that a sufficiently small ε can satisfy the constraints on δ, δ_F in all preceding lemmas. Then, by applying Lemma 8, we get the conclusion.

□

Lemma 12. *Suppose Assumptions 1 and 2 hold, and*

$$\delta_{F_s} \leq \min\left\{\frac{\alpha\beta_2\lambda_{H_{F,s}}}{2}, \frac{\lambda_{H_{F,s}}}{4}, \frac{\alpha}{2\chi\eta_0}\left(\frac{\kappa_F^{1/2}}{\alpha} + \beta_2\right)^{-1}\right\}. \tag{70}$$

Then the trajectory z_t in

$$\begin{cases} \dot{z}_t = -\eta_t \chi P_{\mathcal{R}}(z_t) F(z_t)^{-1}(s_t + \beta_2 \nabla f(z_t)), \\ \dot{s}_t = -\alpha s_t + \nabla f(z_t), \end{cases}$$

starting from $z_t|_{t=0} = z_0$ and $s_0 = 0$ satisfies

$$\int_0^t \eta_t \|P_{\mathcal{R}}(z_t) F(z_t)^{-1} \nabla f(z_t)\|_{F(z_t)}^2 \leq \frac{2(f(z_0) - \inf f)}{\chi\beta_2}, \quad \int_0^t \eta_t \|s_t\|_{F(z_t)}^2 \leq \frac{2(f(z_0) - \inf f)}{\chi\alpha}.$$

Proof. Consider the Lyapunov function $W_t = f(z_t) - \inf f + \frac{1}{2}\eta_t\chi\|s_t\|_{F^{-1}(z_t)}^2$. Then taking the time derivative gives

$$\begin{aligned}\frac{d}{dt}W_t = &\underbrace{-\eta_t\chi\left\langle (P_{\mathcal{R}}(z_t) - I) F(z_t)^{-1}\nabla f(z_t), s_t\right\rangle}_{A} - \eta_t\chi\beta_2 \|P_{\mathcal{R}}(z_t) F(z_t)^{-1}\nabla f(z_t)\|_{F(z_t)}^2 \\ &- \eta_t\chi\alpha\|s_t\|_{F^{-1}(z_t)}^2 + \frac{1}{2}\eta_t\chi s_t^\top \nabla F^{-1}(z_t)[\dot{z}_t] s_t + \frac{1}{2}\dot{\eta}_t\chi\|s_t\|_{F^{-1}(z_t)}^2.\end{aligned} \tag{71}$$

By using $F(z_t)^{-1}\nabla f(z_t) = P_f(z_t) F(z_t)^{-1}\nabla f(z_t)$ and Lemma 8, we have

$$\begin{aligned}\|(P_{\mathcal{R}}(z_t) - I) F(z_t)^{-1}\nabla f(z_t)\|_{F(z_t)}^2 &= \|(P_{\mathcal{R}}(z_t) - P_f(z_t)) F(z_t)^{-1}\nabla f(z_t)\|_{F(z_t)}^2 \\ &\leq \frac{\delta_{F_s}^2}{\lambda_{H_{F,s}}^2} \|F(z_t)^{-1}\nabla f(z_t)\|_{F(z_t)}^2.\end{aligned}$$

Since $\|(P_{\mathcal{R}}(z_t)-I)F(z_t)^{-1}\nabla f(z_t)\|^2_{F(z_t)}+\|P_{\mathcal{R}}F(z_t)^{-1}\nabla f(z_t)\|^2_{F(z_t)} = \|F(z_t)^{-1}\nabla f(z_t)\|^2_{F(z_t)}$, we have

$$\|(P_{\mathcal{R}}(z_t) - I)F(z_t)^{-1}\nabla f(z_t)\|^2_{F(z_t)} \leq \frac{\frac{\delta^2_{F_s}}{\lambda^2_{H_{F,s}}}}{1-\frac{\delta^2_{F_s}}{\lambda^2_{H_{F,s}}}}\|P_{\mathcal{R}}(z_t)F(z_t)^{-1}\nabla f(z_t)\|^2_{F(z_t)}.$$

Thus

$$\begin{aligned} A \overset{(70)}{\leq} & 2\frac{\delta_{F_s}}{\lambda_{H_{F,s}}}\eta_t\chi\|P_{\mathcal{R}}(z_t)F(z_t)^{-1}\nabla f(z_t)\|_{F(z_t)}\|s_t\|_{F^{-1}(z_t)} \\ \leq & \frac{\delta_{F_s}}{\alpha\lambda_{H_{F,s}}}\eta_t\chi\|P_{\mathcal{R}}(z_t)F(z_t)^{-1}\nabla f(z_t)\|^2_{F(z_t)} + \frac{\delta_{F_s}\alpha}{\lambda_{H_{F,s}}}\eta_t\chi\|s_t\|^2_{F^{-1}(z_t)}. \end{aligned} \tag{72}$$

On the other hand, similar to (47), we have

$$\begin{aligned} \|\dot{z}_t\|_{F(z_t)} \leq & \eta_t\chi\left(\|s_t\|_{F^{-1}(w_t)} + \beta_2\|\nabla f(z_t)\|_{F^{-1}(z_t)}\right) \\ \leq & \eta_t\chi\left(\lambda_F^{-\frac{1}{2}}\int_0^t e^{\alpha-(t-s)}\|\nabla f(z_s)\|_2 ds + \beta_2\|\nabla f(z_t)\|_{F^{-1}(z_t)}\right) \\ \leq & \eta_t\,\underbrace{\chi\left(\frac{\kappa_F^{1/2}}{\alpha}+\beta_2\right)}_{:=C_w}G. \end{aligned}$$

This gives

$$\eta_t\chi s_t^\top\nabla F^{-1}(z_t)[\dot{z}_t]s_t \leq \eta_t^2\chi\delta_{F_s}C_w\|s_t\|^2_{F^{-1}(z_t)} \overset{(70)}{\leq} \frac{1}{2}\eta_t\chi\alpha\|s_t\|^2_{F^{-1}(z_t)}. \tag{73}$$

Substituting (72) and (73) into (71) yields

$$\frac{d}{dt}W_t = -\frac{1}{2}\eta_t\chi\beta_2\|P_{\mathcal{R}}(z_t)F(z_t)^{-1}\nabla f(z_t)\|^2_{F(z_t)} - \frac{1}{2}\eta_t\chi\alpha\|s_t\|^2_{F^{-1}(z_t)}. \tag{74}$$

where we also uses $\dot{\eta}_t \leq 0$. Finally, integrating both sides of (74) from 0 to t yields the conclusion.

□

J.3 Useful Lemmas

Lemma 13 (Cauchy's Inequality)**.** *For any vector u, v and positive semi-definite matrices A and F, by using $u^\top Av \leq \|A^{\frac{1}{2}}u\|_2\|A^{\frac{1}{2}}v\|_2 \leq \|A\|_2\|u\|_2\|v\|_2$, we have*

$$u^\top Av = u^\top F^{\frac{1}{2}}F^{-\frac{1}{2}}AF^{-\frac{1}{2}}F^{\frac{1}{2}}v \leq \|F^{-\frac{1}{2}}AF^{-\frac{1}{2}}\|_2\|u\|_F\|v\|_F.$$

Lemma 14. *Let $a_t, b_t \geq 0$ be real-valued functions defined for $t \geq 0$. Suppose a_t is continuously differentiable and b_t is continuous with respect to t. If there exists a constant $\lambda > 0$ such that*

$$\frac{d}{dt}a_t \leq -\lambda a_t + \sqrt{a_t}\sqrt{b_t}, \tag{75}$$

then for all $t \geq 0$,

$$\sqrt{a_t} \leq e^{-\frac{\lambda}{2}t}\sqrt{a_0} + \frac{1}{2}\int_0^t e^{-\frac{\lambda}{2}(t-s)}\sqrt{b_s}ds.$$

Proof. Let $\delta > 0$ be an arbitrary small constant. Define a perturbed variable y_t as $y_t = \sqrt{a_t+\delta}$. Since $a_t \geq 0$, we have $y_t \geq \sqrt{\delta} > 0$ for all t. Thus, y_t is continuously differentiable. Substituting $\frac{d}{dt}y_t$ into (75) gives

$$\frac{d}{dt}y_t \leq -\frac{\lambda}{2}y_t + \frac{\lambda\delta}{2y_t} + \frac{\sqrt{b_t}}{2}\frac{\sqrt{y_t^2-\delta}}{y_t} \leq -\frac{\lambda}{2}y_t + \frac{\sqrt{b_t}}{2} + \frac{\lambda\sqrt{\delta}}{2}.$$

It follows that

$$\frac{d}{dt}\left(e^{\frac{\lambda}{2}t}y_t\right) \le e^{\frac{\lambda}{2}t}\left(\frac{\sqrt{b_t}}{2} + \frac{\lambda\sqrt{\delta}}{2}\right).$$

Integrating from 0 to t yields:

$$y_t \le e^{-\frac{\lambda}{2}t}y_0 + \frac{1}{2}\int_0^t e^{-\frac{\lambda}{2}(t-s)}\sqrt{b_s}\,ds + \frac{\lambda\sqrt{\delta}}{2}\int_0^t e^{-\frac{\lambda}{2}(t-s)}ds.$$

Take the limit as $\delta \to 0$, we get the conclusion.

□

Remark 8. *It seems natural to divide the inequality (75) by $\sqrt{a_t}$ and consider $\frac{d}{dt}\sqrt{a_t}$. However, direct differentiation of $\sqrt{a_t}$ is not well-defined when $a_t = 0$. To handle this singularity strictly, we introduce a regularization term δ.*

Lemma 15 (Gronwall's Inequality)**.** *Let u_t, α_t and $\beta_t \ge 0$ be continuous functions for $t \ge 0$. If*

$$u_t \le \alpha_t + \int_0^t \beta_s u_s ds,$$

then

$$u_t \le \alpha_t + \int_0^t \alpha_s\beta_s \exp\left(\int_s^t \beta_\tau d\tau\right) ds.$$

Additionally, if α_t is non-decreasing, then

$$u_t \le \alpha_t \exp\left(\int_0^t \beta_\tau d\tau\right).$$

Lemma 16 (Wedin $\sin(\Theta)$ Theorem)**.** *Let M and N be two matrices with singular values $\sigma_1, \dots, \sigma_n$ and $\tilde{\sigma}_1, \dots, \tilde{\sigma}_n$, respectively. Denote by U_M, V_M the first k left and right singular vectors of M, and by U_N, V_N those of N. Suppose there exists $\alpha > 0$ such that*

$$\alpha \le \min_{1\le i\le k}\sigma_i - \max_{k+1\le r\le n}\tilde{\sigma}_r.$$

Denote the projection to the subspace spanned by U_M as $\mathcal{P}_{U_M}$ (and similarly for other matrices). Then

$$\max\left\{\|\mathcal{P}_{U_M} - \mathcal{P}_{U_N}\|_2,\ \|\mathcal{P}_{V_M} - \mathcal{P}_{V_N}\|_2\right\} \le \frac{\|M-N\|_2}{\alpha}.$$

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We believe that the main claims in the abstract and introduction accurately reflect the contributions and scope of 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: The limitations are in Section Conclusion.

   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: See Section 7 and Appendix J.

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: We have explained our method and provided experimental details in Sections 5, 6 and Appendices B, C.

   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: The dataset we used is public and can be accessed by everyone. We have provided the code that reproduces the main experimental results.

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: The details are in Appendices B, C.

   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: We report the results in Appendix D.3.

   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: The details are in Appendix C.2.

   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: We have confirmed that the research is conducted with the NeurIPS 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: [N/A]

    Justification: There is no societal impact of our work performed.

    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: The paper poses no such risks.

    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: We have cited the original papers that produced the code package or dataset.

    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: The 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: The paper does not involve crowdsourcing nor research 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: The paper does not involve crowdsourcing nor research 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: LLMs are not used in the core methodology, experimental design, or result analysis.

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.