# BREAKING THE BLACK BOX: AN ARCHITECTURE-SPECIALIZED PRECONDITIONING OPTIMIZER FOR ATTENTION MECHANISM

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

The increasing scale of deep learning models underscores the importance of efficient optimization. Existing matrix-based optimizers exploit parameter matrix structure through gradient statistics, activation statistics, or matrix transformations, but generally do not explicitly incorporate the analytical dependencies induced by specific network architectures. In this work, we propose STRATA, a structure-aware preconditioning optimizer for Transformer training. Moving from a black-box to a "gray-box" perspective, we explicitly exploit the coupling between the Query and Key projections in Attention to identify an architecture-specific structural curvature factor. We develop a two-stage approximation that transforms this activation-dependent quantity into a practical preconditioner constructed only from readily available model parameters and parameter gradients. We further characterize the approximation errors introduced by the two stages under a stylized model of the Attention distribution. Empirically, we evaluate STRATA on LLaMA-style pre-training from 130M to 1.3B parameters. Across the tested model scales, STRATA consistently improves optimization efficiency over AdamW and Muon and achieves end-to-end wall-clock speedups of 11%–31% relative to Muon, with negligible additional peak-memory overhead.

## 1 INTRODUCTION

Gradient-based optimization is a central component of large-scale deep learning (Bottou et al., 2018). As model and data scales continue to increase following neural scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022), the computational cost of training has become a major bottleneck (Shoeybi et al., 2019; Narayanan et al., 2021). Improving the optimization efficiency of large neural networks is therefore increasingly important, especially for Transformer-based models whose training requires substantial computational resources.

A broad line of work improves optimization through preconditioning. Vector-based methods such as Adam and AdamW (Kingma, 2014; Loshchilov & Hutter, 2017) construct coordinate-wise adaptive updates, while matrix-based methods exploit the matrix structure of neural network parameters. Representative examples include Shampoo (Gupta et al., 2018), which constructs Kronecker-structured matrix preconditioners from accumulated gradient statistics, and Muon (Jordan et al., 2024), which applies matrix orthogonalization and has recently demonstrated strong performance in large-scale language model training (Liu et al., 2025a). Despite their different constructions, these methods primarily build preconditioners from generic local statistics such as parameter gradients or activation-gradient covariances. The internal architecture that generates these statistics is typically not explicitly incorporated into the optimizer design.

However, modern Transformer models are not arbitrary collections of matrix parameters. The Attention mechanism contains explicit algebraic interactions among the Query, Key, and Value projections, together with a structured softmax transformation (Vaswani et al., 2017). These architectural relations induce parameter interactions that are invisible when each matrix is optimized only through its own gradient statistics. This motivates an intermediate perspective between treating the objective as a complete black box and attempting to analyze the full network as a white box. We refer to



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

this perspective as a *gray-box* view: rather than exposing the entire model, we selectively exploit tractable architectural components whose structure can directly inform optimizer construction.

Following this principle, we study the curvature structure induced by the Attention mechanism and develop STRATA, a structure-aware preconditioning optimizer for Transformer training. We first derive an Attention-specific structural curvature factor arising from the coupling between the Query and Key projections. We then construct a two-stage approximation that transforms this activation-dependent quantity into an activation-free surrogate using the Value parameter and its gradient. This construction produces a practical right-sided preconditioner for the Query and Key projections that depends only on readily available parameters and parameter gradients, while the remaining model parameters can be updated by a standard base optimizer.

**Contribution.** We address the above challenge in three steps. We first identify the architecture-specific curvature induced by the Query-Key coupling in Attention. We then develop an activation-free approximation of this structure and convert it into a practical right-sided preconditioner, leading to STRATA. Finally, we evaluate the resulting optimizer across multiple Transformer scales and demonstrate its practical training efficiency.

First, we introduce a *gray-box optimization perspective* that explicitly exploits the internal structure of Attention rather than treating the model as an architecture-agnostic objective. Starting from the parameter-space curvature of the Query projection, we show that its generalized Gauss-Newton matrix admits a bilateral structure consisting of an input-dependent left factor and an Attention-specific right factor. The latter directly depends on the Key representation and the softmax-gradient geometry, thereby exposing the structural coupling between the Query and Key projections. The same construction applies symmetrically to the Key projection.

Second, we develop a *two-stage activation-free approximation* that turns this structural factor into a practical optimizer. The first stage simplifies the softmax-gradient interaction, while the second removes the remaining activation dependence through the Value projection and its gradient. Let $\mathbf{W}_V$ and $\mathbf{W}_K$ denote the Value and Key projection matrices, respectively, and let $\mathbf{G}_V := \nabla\mathcal{L}(\mathbf{W}_V)$ denote the Value gradient. STRATA constructs $\mathbf{U}_Q = \mathbf{W}_V\mathbf{G}_V^\top\mathbf{W}_K$ and $\mathbf{H}_Q = \mathbf{U}_Q^\top\mathbf{U}_Q + \epsilon\mathbf{I}$, where $\epsilon > 0$ is the damping parameter and $\mathbf{I}$ is the identity matrix, and uses $\mathbf{H}_Q$ as the right preconditioner for the Query update. The Key update follows symmetrically. We additionally characterize the approximation errors introduced by the two stages under the stated assumptions, providing an explanatory view of when the proposed surrogate better reflects the underlying structural curvature.

Finally, we evaluate STRATA on LLaMA-style pre-training from 130M to 1.3B parameters against AdamW and Muon. Across the tested model scales, STRATA consistently improves optimization efficiency and achieves end-to-end wall-clock speedups of $11\%$–$31\%$ for reaching the corresponding loss targets, with negligible additional peak-memory overhead.

### 1.1 Related Work

**Matrix-Based Optimization.** Matrix-based optimization exploits the matrix structure of neural network parameters to construct updates beyond coordinate-wise scaling. A representative example is K-FAC (Martens & Grosse, 2015), which approximates the Fisher information matrix through Kronecker factors formed from activation and backpropagated-gradient statistics. Shampoo (Gupta et al., 2018) instead constructs left- and right-sided preconditioners from accumulated parameter gradients and motivates the resulting matrix update through a Kronecker-structured upper bound. More recently, Muon (Jordan et al., 2024; Liu et al., 2025a) applies matrix orthogonalization to the gradient and uses efficient Newton–Schulz iterations to approximate the corresponding transformation, demonstrating strong empirical performance in large-scale language model training. Other examples include SOAP (Vyas et al., 2024), COSMOS (Liu et al., 2025b), AdaMuon (Si et al., 2025), and PolarGrad (Lau et al., 2025). Although these methods exploit matrix-form parameter structure, their preconditioners are constructed from different sources of information and generally remain architecture-agnostic. In contrast, our work explicitly leverages the analytical coupling among parameter matrices within Transformer Attention to construct an architecture-specific preconditioner.

**Architecture-Aware Optimization.** A growing line of theoretical work exploits explicit neural architectures to better understand their optimization dynamics. For Transformer models, Tian et al. (2023) study the training dynamics and token composition of a one-layer Transformer, while Li

et al. (2024) analyze the optimization and generalization behavior of two-layer Transformers under sign gradient descent. These studies demonstrate that exposing the internal Attention structure can reveal optimization properties that are difficult to capture under a fully architecture-agnostic view. However, such analyses primarily focus on simplified models, and translating architectural structure into practical optimizers for large-scale Transformer training remains comparatively underexplored.

Related ideas also appear in structured optimization. In low-rank matrix estimation and matrix sensing, the coupled structure between factorized variables has been explicitly exploited to improve conditioning and accelerate optimization (Tong et al., 2021; Xu et al., 2023). These results suggest that analytical dependencies among parameter blocks can provide the approximation of higher-order precondition beyond their individual gradients. Inspired by this perspective, our work exploits the explicit Query-Key coupling in Attention and uses it directly to construct a structure-aware preconditioner for Transformer pre-training.

## 2 Structured Curvature and Matrix Preconditioning

In this section, we introduce a common curvature-based view of matrix preconditioning that will be used to position existing matrix optimizers and motivate our subsequent structure-aware design. Consider a matrix parameter $\mathbf{W} \in \mathbb{R}^{D\times d}$. Second-order curvature with respect to $\mathbf{W}$ is naturally represented in the vectorized parameter space, where $\mathrm{vec}(\mathbf{W}) \in \mathbb{R}^{Dd}$. The corresponding curvature matrix therefore has size $Dd \times Dd$, leading to a memory cost of $\mathcal{O}(D^2d^2)$ if represented explicitly. This makes direct preconditioning with the full curvature matrix impractical for large-scale neural network training. A natural alternative is to seek a structured approximation that preserves interactions among parameter coordinates while admitting a tractable representation. In particular, suppose the curvature matrix, or an approximation to it, admits the Kronecker-structured form $\mathbf{B}^\top \otimes \mathbf{A}$, where $\mathbf{A} \in \mathbb{R}^{D\times D}$ and $\mathbf{B} \in \mathbb{R}^{d\times d}$. Its action on a vectorized matrix is equivalently represented by

$$\left(\mathbf{B}^\top \otimes \mathbf{A}\right) \mathrm{vec}(\mathbf{C}) = \mathrm{vec}\left(\mathbf{ACB}\right). \tag{1}$$

Therefore, a Kronecker-structured curvature approximation converts a high-dimensional operator in the vectorized parameter space into left- and right-sided transformations of the original matrix. This reduces the storage cost from $\mathcal{O}(D^2d^2)$ to $\mathcal{O}(D^2 + d^2)$ and provides a tractable route from second-order curvature information to matrix-form preconditioning.

This curvature-based view provides a common lens for relating several representative matrix optimizers, although their preconditioners are constructed from different sources of information. K-FAC (Martens & Grosse, 2015) directly exploits a Kronecker-factored approximation of the Fisher information matrix, with the two factors constructed from activation statistics and backpropagated gradients. This provides a principled curvature approximation, but requires additional collection and maintenance of activation-dependent statistics. Shampoo (Gupta et al., 2018) instead constructs matrix preconditioners from accumulated parameter gradients and establishes a Kronecker-structured upper bound on the corresponding second-order matrix. Its practical implementation, however, requires computing fractional matrix powers, which can introduce non-negligible computational overhead at large scale. More recently, Muon (Jordan et al., 2024) applies matrix orthogonalization to gradient updates and uses Newton-Schulz iterations to compute the transformation efficiently. Although Muon is not originally derived from an explicit curvature factorization, recent work (Du & Su, 2026) shows that, under additional local approximation and activation assumptions, its update admits a simplified Newton-type interpretation involving Kronecker-structured parameter-space curvature. Therefore, these methods can be related through the structured-curvature view introduced above, while differing substantially in how the matrix preconditioners are constructed. We provide further details on these connections in Appendix A.

The discussion above concerns how high-dimensional curvature information can be represented and applied tractably to matrix parameters. A separate design question is where the information used to construct such a preconditioner should come from. Existing matrix optimizers typically build their preconditioners from activation statistics, parameter gradients, or transformations of the gradients themselves. These constructions are broadly applicable across different network architectures, but they do not explicitly exploit analytical dependencies among distinct parameter matrices within a specific computational module. However, Transformer Attention provides such a dependency. In particular, the Query and Key projections $\mathbf{W}_Q$ and $\mathbf{W}_K$ jointly determine the pre-softmax Attention scores through their bilinear interaction. This coupling implies that the local curvature with respect

to one projection depends structurally on the other projection. Therefore, rather than constructing the preconditioner solely from parameter-local statistics, we ask whether this architecture-specific coupling can be used to identify a structural curvature factor for the Query and Key parameters. This perspective complements the curvature-based matrix preconditioning view above by changing the source of the preconditioning, from generic local statistics to explicit cross-parameter structure.

# 3 Derivation: Towards a Gray Box Model

In this section, we first propose an objective function for an explicitly parameterized Attention mechanism, which can be regarded as a **gray box** model. Following a structural analysis of the second-order information of this objective function, we construct the preconditioner with a controllable error to approximate the Newton update in matrix form.

## 3.1 Attention-Specific Structural Curvature

In this section, we address the following question: *what curvature structure does the Attention mechanism induce for the Query and Key parameters?* To answer this question, we explicitly expose the Attention computation while leaving the remaining network components unspecified. This gray-box formulation allows us to identify an Attention-specific curvature target before introducing the practical approximations.

Specifically, we consider a learning setup with a dataset $\mathcal{D}$ consisting of a series of i.i.d. context samples $\{\mathbf{X}, \mathbf{y}\}$, where $\mathbf{X} \in \mathbb{R}^{R\times D}$, with $R$ denoting the sequence length and $D$ denoting the hidden size, and $\mathbf{y} \in \mathbb{R}^{C}$. A generic formulation typically represents the neural network as $f(\mathbf{W};\cdot) : \mathbb{R}^{R\times D} \to \mathbb{R}^{C}$, parameterized by an architecture-agnostic parameter collection $\mathbf{W}$, thereby treating the internal network structure as a black box. In contrast, our gray-box formulation explicitly exposes the Attention mechanism and writes $f(\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V; \mathbf{X}) = f(\mathbf{Z})$, where

$$\mathbf{Z} = \mathrm{softmax}_{\mathrm{row}}\left(\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top\right)\mathbf{X}\mathbf{W}_V, \tag{2}$$

with $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{D\times d}$ denoting the Query, Key, and Value projection matrices, respectively. The operator $\mathrm{softmax}_{\mathrm{row}}(\cdot)$ applies the softmax function $\mathrm{softmax}(\mathbf{x})_i := \exp(x_i)/\sum_{j=1}^{J}\exp(x_j)$, for an input $\mathbf{x} \in \mathbb{R}^{J}$, independently to each row of the input matrix. Without loss of generality, we consider single-head Attention. It is worth noting that $d \le D$, since in a multi-head Attention architecture, the concatenation of the head dimension $d$ across all heads typically recovers the total hidden dimension $D$.

In a multi-layer architecture, $\mathbf{X}$ in (2) refers to the intermediate representation generated by propagating the raw context sample through the preceding layers. Therefore, we use $\mathbf{X}$ to denote the local input to the current Attention layer, whether it corresponds to the original context sample or an intermediate activation. The two weight matrices $\mathbf{W}_Q$ and $\mathbf{W}_K$ are coupled through the Attention logits and are applied to all $R$ tokens. Except for the explicitly exposed Attention mechanism above, other structures, such as MLP layers, normalization layers, and their corresponding parameters, remain hidden in $f$. Our objective is to minimize the loss function associated with this gray-box model, $\mathcal{L}\left(f(\mathbf{Z}), \mathbf{y}\right) : \mathbb{R}^{C}\times\mathbb{R}^{C}\to\mathbb{R}$.

We first analyze the curvature with respect to $\mathbf{W}_Q$. The corresponding result for $\mathbf{W}_K$ follows from the symmetry between the Query and Key projections. Define the Key representation $\mathbf{K} = \mathbf{X}\mathbf{W}_K$ and the pre-softmax Attention logits $\mathbf{A} = \mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top$. Since $\mathbf{A}$ is linear in $\mathbf{W}_Q$ when $\mathbf{X}$ and $\mathbf{W}_K$ are fixed, the exact parameter-space Hessian with respect to $\mathrm{vec}(\mathbf{W}_Q)$ can be written as

$$\mathcal{H}_{\mathrm{vec}(\mathbf{W}_Q)} = \left(\mathbf{K}\otimes\mathbf{X}\right)^\top\nabla^2_{\mathrm{vec}(\mathbf{A})}\mathcal{L}\left(\mathbf{K}\otimes\mathbf{X}\right). \tag{3}$$

This representation makes the Query-Key coupling explicit: the parameter-space curvature with respect to $\mathbf{W}_Q$ depends directly on the Key representation $\mathbf{K}$, and analogously the curvature with respect to $\mathbf{W}_K$ depends on the Query representation.

The exact Hessian in (3) is generally indefinite and remains difficult to characterize for the full network. We therefore use the generalized Gauss-Newton matrix as a positive semidefinite curvature surrogate and study its structure under the gray-box model. The following lemma identifies the corresponding Kronecker factors.

**Lemma 3.1.** *Suppose $\mathcal{L}$ is the cross-entropy loss and let $\tilde{\mathcal{H}}_{\mathrm{vec}(\mathbf{W}_Q)} = \mathbb{E}[\mathcal{H}_{\mathrm{vec}(\mathbf{W}_Q)}]$ be the expected generalized Gauss-Newton matrix of the Query parameter $\mathbf{W}_Q$. Denoting the gradient with respect to the softmax logits $\mathbf{A}$ as $\mathbf{G} \in \mathbb{R}^{R\times R}$ such that $\mathrm{vec}(\mathbf{G}) = \frac{\partial \mathcal{L}(f(\mathbf{Z}),y)}{\partial\, \mathrm{vec}(\mathbf{A})}$, and assuming statistical independence between the structural inputs and the gradient covariance, the expected generalized Gauss-Newton matrix can be Kronecker-factored as:*

$$\tilde{\mathcal{H}}_{\mathrm{vec}(\mathbf{W}_Q)} \approx \mathbb{E}\left[\mathbf{K}^\top \mathbf{G}^\top \mathbf{G}\mathbf{K}\right] \otimes \mathbb{E}\left[\mathbf{X}^\top \mathbf{X}\right]. \tag{4}$$

*For any preconditioning order $\gamma \in (0, 1]$, applying this preconditioner to the vectorized gradient $\mathrm{vec}(\nabla\mathcal{L}(\mathbf{W}_Q))$ yields the matrix-form preconditioned update:*

$$\left(\tilde{\mathcal{H}}_{\mathrm{vec}(\mathbf{W}_Q)}\right)^{-\gamma} \mathrm{vec}(\nabla\mathcal{L}(\mathbf{W}_Q)) = \mathrm{vec}\left(\left(\mathbb{E}\left[\mathbf{X}^\top\mathbf{X}\right]\right)^{-\gamma} \nabla\mathcal{L}(\mathbf{W}_Q) \left(\mathbb{E}\left[\mathbf{K}^\top\mathbf{G}^\top\mathbf{G}\mathbf{K}\right]\right)^{-\gamma}\right). \tag{5}$$

The proof is provided in Appendix D. Lemma 3.1 identifies two distinct curvature factors for the Query parameter. The left factor $\mathbb{E}[\mathbf{X}^\top\mathbf{X}]$ captures the covariance structure of the local input, whereas the right factor $\mathbb{E}[\mathbf{K}^\top\mathbf{G}^\top\mathbf{G}\mathbf{K}]$ explicitly incorporates the Key representation and therefore reflects the Query-Key coupling induced by Attention. The same argument yields the symmetric counterpart for $\mathbf{W}_K$.

### 3.2 Efficient Approximation of the Structural Curvature

In this section, we address the following question: *how can the Attention-specific structural curvature identified above be approximated without accessing activations?* Specifically, we aim to construct an activation-free approximation of the right curvature factor $\mathbf{H} \approx \mathbb{E}\left[\mathbf{K}^\top\mathbf{G}^\top\mathbf{G}\mathbf{K}\right]$ using only readily available parameters and their gradients. We achieve this through two successive approximations. First, we exploit the analytical structure of the softmax gradient and decompose the logit gradient $\mathbf{G}$ into a main term and a normalization term. We then replace $\mathbf{G}$ by its main term, reducing the original curvature factor to a simpler intermediate target, and quantify the approximation gap introduced by this replacement. Second, although this intermediate target is simpler, it still depends explicitly on the local input $\mathbf{X}$. We therefore exploit the algebraic structure of the Value gradient $\nabla\mathcal{L}(\mathbf{W}_V)$ to construct a surrogate using only $\mathbf{W}_V$, $\nabla\mathcal{L}(\mathbf{W}_V)$, and the Key parameter $\mathbf{W}_K$, and quantify the additional gap introduced by eliminating the activation-dependent interaction. These two steps transform the structural curvature identified in Section 3.1 into a practical activation-free approximation, while separately characterizing the error introduced at each stage.

**Stage I. Softmax Gradient Decomposition.** We first simplify the dependence of the structural curvature factor on the logit gradient $\mathbf{G}$. Since $\mathbf{K} = \mathbf{X}\mathbf{W}_K$, we have $\mathbf{K}^\top\mathbf{G}^\top\mathbf{G}\mathbf{K} = \mathbf{W}_K^\top\mathbf{X}^\top\mathbf{G}^\top\mathbf{G}\mathbf{X}\mathbf{W}_K$. The Key parameter $\mathbf{W}_K$ is directly available during optimization, whereas the remaining difficulty lies in the structure of $\mathbf{G} = \frac{\partial\mathcal{L}}{\partial\mathbf{A}}$. We therefore begin by characterizing $\mathbf{G}$ through the softmax Jacobian. The following lemma provides an exact decomposition of the logit gradient into a main term and a normalization term.

**Lemma 3.2.** *For $\mathbf{G} = \frac{\partial\mathcal{L}}{\partial\mathbf{A}}$ where $\mathcal{L}(f(\mathbf{Z}), \mathbf{y})$ is the loss function with $\mathbf{Z}$ is the Attention output defined as* (2)*, $\mathbf{S}$ is the softmax output in Attention structure, and $\mathbf{A}$ is the softmax logits of $\mathbf{S}$, we have*

$$\mathbf{G} = \underbrace{\left(\frac{\partial\mathcal{L}}{\partial\mathbf{Z}}\mathbf{W}_V^\top\mathbf{X}^\top\right)\odot\mathbf{S}}_{\text{main}} - \underbrace{\mathbf{D}\mathbf{S}}_{\text{bias}}, \tag{6}$$

*where $\mathbf{D} = \frac{\partial\mathcal{L}}{\partial\mathbf{Z}}\mathbf{W}_V^\top\mathbf{X}^\top\mathbf{S}^\top \odot \mathbf{I}$ is a diagonal matrix.*

Lemma 3.2 gives an exact decomposition of $\mathbf{G}$. To obtain a simpler intermediate curvature target, we retain the main term and use it as an approximation of $\mathbf{G}$. This constitutes the first approximation in our construction. The following theorem quantifies the discrepancy introduced by this replacement under a stylized model of the softmax distribution.

**Theorem 3.3.** *Suppose the distribution of $\mathbf{S} \in \mathbb{R}^{R\times R}$ satisfies that for each row $\mathbf{s}_i$, there exists $0 < p_0 = \mathcal{O}\left(R^{-\beta}\right), \beta \geq 1$, and a positive integer $k$, such that $R - k$ dimensions of $\mathbf{s}_i$ hold the*

*probability $p_0$, and other $k$ dimensions hold the probability $p_1 = \frac{1-p_0 R}{k} + p_0$ with $k = \Theta\left(R^{\alpha}\right), \alpha \in [0,1]$. Denoting $\mathbf{G} = \mathbf{M} - \mathbf{N}$, where $\mathbf{M} = \bar{\mathbf{S}} \odot \mathbf{S}, \mathbf{N} = \left(\bar{\mathbf{S}}\mathbf{S}^\top \odot \mathbf{I}\right)\mathbf{S}, \bar{\mathbf{S}} = \frac{\partial \mathcal{L}}{\partial \mathbf{Z}} \mathbf{W}_V^\top \mathbf{X}^\top$, we have*

$$\|\mathbf{M} - \mathbf{G}\|_F \leq \mathcal{O}\left(R^{\frac{1-\alpha}{2}}\right) \|\mathbf{M}\|_F. \tag{7}$$

Theorem 3.3 characterizes the first approximation gap in our construction, namely the discrepancy introduced by replacing the exact logit gradient $\mathbf{G}$ with its main term $\mathbf{M}$. Under the stated softmax distribution model, this gap is controlled by the sequence length $R$ and the concentration parameter $\alpha$. In particular, for a fixed sequence length, the upper bound decreases as $\alpha$ increases, indicating that this approximation is more accurate for less concentrated softmax distributions under the assumed model. Therefore, Stage I replaces the original factor $\mathbf{K}^\top \mathbf{G}^\top \mathbf{G} \mathbf{K}$ with the simpler intermediate target $\mathbf{K}^\top \mathbf{M}^\top \mathbf{M} \mathbf{K}$, while making the error associated with this replacement explicit. This intermediate target still depends on the local activation $\mathbf{X}$, which motivates the activation-free approximation developed in Stage II.

**Stage II. Activation-Free Approximation.** After Stage I, the intermediate curvature target can be written as $\mathbf{K}^\top \mathbf{M}^\top \mathbf{M} \mathbf{K} = \mathbf{W}_K^\top \mathbf{X}^\top \mathbf{M}^\top \mathbf{M} \mathbf{X} \mathbf{W}_K$. We notice the middle factor $\mathbf{X}^\top \mathbf{M}^\top \mathbf{M} \mathbf{X}$ still depends explicitly on the local activation $\mathbf{X}$ and the softmax output $\mathbf{S}$. Constructing the preconditioner directly from these quantities would therefore require additional access to activation-dependent information during training. To obtain an activation-free approximation, we further exploit the structure of the Value projection.

The gradient with respect to the Value parameter satisfies $\nabla \mathcal{L}(\mathbf{W}_V) = \mathbf{X}^\top \mathbf{S}^\top \frac{\partial \mathcal{L}}{\partial \mathbf{Z}}$. Denoting $\mathbf{Q} = \frac{\partial \mathcal{L}}{\partial \mathbf{Z}} \mathbf{W}_V^\top$, we obtain $\nabla \mathcal{L}(\mathbf{W}_V) \mathbf{W}_V^\top = \mathbf{X}^\top \mathbf{S}^\top \mathbf{Q}$. This relation suggests the activation-free surrogate

$$\mathcal{K}_{\text{approx}} = \mathbf{X}^\top \mathbf{S}^\top \mathbf{Q} \mathbf{Q}^\top \mathbf{S} \mathbf{X} = \nabla \mathcal{L}(\mathbf{W}_V) \mathbf{W}_V^\top \mathbf{W}_V \nabla \mathcal{L}(\mathbf{W}_V)^\top, \tag{8}$$

which can be computed using only the Value parameter and its gradient. Consequently, after incorporating the Key projection, the corresponding right preconditioner takes the form $\mathbf{W}_K^\top \nabla \mathcal{L}(\mathbf{W}_V) \mathbf{W}_V^\top \mathbf{W}_V \nabla \mathcal{L}(\mathbf{W}_V)^\top \mathbf{W}_K$. The following Theorem quantifies the approximation gap between the activation-dependent intermediate target and this activation-free surrogate.

**Theorem 3.4.** *Denote $\mathbf{Q} = \frac{\partial \mathcal{L}}{\partial \mathbf{Z}} \mathbf{W}_V^\top$, $\mathcal{K}_{\text{target}} = \mathbf{X}^\top \mathbf{M}^\top \mathbf{M} \mathbf{X}$, and $\mathcal{K}_{\text{approx}} = \mathbf{X}^\top \mathbf{S}^\top \mathbf{Q} \mathbf{Q}^\top \mathbf{S} \mathbf{X}$. Let $\mathbf{Q} = [\mathbf{q}_1^\top, \ldots, \mathbf{q}_R^\top]^\top$ and $\mathbf{X} = [\mathbf{x}_1^\top, \ldots, \mathbf{x}_R^\top]^\top$. Suppose that: (1) the rows of $\mathbf{Q}$ satisfy $\mathbb{E}[\mathbf{q}_i \mathbf{q}_j^\top] = \mathbf{0}$ for all $i, j \in [R]$ with $i \neq j$; (2) $\|\mathbf{Q}\|_F \leq D_0$; (3) the local inputs satisfy $\mathbb{E}[\mathbf{x}_i \mathbf{x}_i^\top] = \mathbf{I}$ for all $i \in [R]$ and $\mathbb{E}[\mathbf{x}_i \mathbf{x}_j^\top] = \mathbf{0}$ for all $i, j \in [R]$ with $i \neq j$; (4) each entry of $\mathbf{X}$ is bounded as $|[\mathbf{X}]_{ij}| \leq \delta$, and $\mathbf{X}$ is statistically independent of the other random variables involved in the expectation; and (5) the distribution of $\mathbf{S} \in \mathbb{R}^{R \times R}$ satisfies that, for each row $\mathbf{s}_i$, there exists $0 < p_0 = \mathcal{O}(R^{-\beta})$, with $\beta \geq 1$, and a positive integer $k$ such that $R - k$ entries of $\mathbf{s}_i$ equal $p_0$, while the remaining $k$ entries equal $p_1 = \frac{1-p_0 R}{k} + p_0$, where $k = \Theta(R^{\alpha})$ and $\alpha \in [0,1]$. Then,*

$$\mathbb{E}\|\Delta\|_F \leq \mathcal{O}\left(\delta^2 D_0 D^2 R^{\frac{1}{2}} \left(1 - R^{-\alpha}\right)\right), \tag{9}$$

*where $\Delta = \mathcal{K}_{\text{target}} - \mathcal{K}_{\text{approx}}$.*

Theorem 3.4 characterizes the second approximation gap in our construction, namely the discrepancy introduced by replacing the activation-dependent target $\mathcal{K}_{\text{target}}$ with the activation-free surrogate $\mathcal{K}_{\text{approx}}$. Under the stated assumptions, the upper bound depends on the softmax concentration parameter $\alpha$ through $1 - R^{-\alpha}$. For a fixed sequence length, this bound decreases as $\alpha$ decreases, indicating that the second approximation becomes more accurate for more concentrated softmax distributions under the assumed model. Together with the result of Stage I, this establishes the two approximation steps required to transform the Attention-specific structural curvature into a preconditioner that can be constructed from readily available parameters and gradients.

The two approximation bounds reveal a complementary dependence on the softmax concentration parameter $\alpha$. For a fixed sequence length $R$, the bound in Theorem 3.3 decreases as $\alpha$ increases, indicating that the approximation $\mathbf{G} \approx \mathbf{M}$ is more accurate for less concentrated softmax distributions under the stated model. In contrast, the bound in Theorem 3.4 decreases as $\alpha$ decreases, indicating that the activation-free approximation of $\mathcal{K}_{\text{target}}$ becomes more accurate for more concentrated softmax distributions. Therefore, the two approximation steps exhibit opposite monotonic dependence

**Algorithm 1** STRATA: STructure-aware pReconditioning for Accelerated TrAnsformers optimization

**Require:** parameters $\mathbf{W} \in \{\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{D\times d}\} \cup \{\mathbf{W}_{\text{others}}\}$, loss function $\mathcal{L} : \mathbb{R}^C \times \mathbb{R}^C \to \mathbb{R}$, dataset $\mathcal{D}$, damping $\epsilon$, learning rate $\eta$, training steps $T$, optimizer for others $\mathcal{A}$, learning rate for others $\eta_{\text{others}}$, precondition order $\gamma \in (0, 1]$

**while** $t < T$ **do**
  **if** $\mathbf{W} \in \{\mathbf{W}_Q, \mathbf{W}_K\}$ **then**
    **Step I. Construct preconditioners for** $\mathbf{W}_Q.\mathbf{W}_K$
    $\mathbf{U}_Q^t = \mathbf{W}_V^t \nabla\mathcal{L} \left(\mathbf{W}_V^t\right)^\top \mathbf{W}_K^t$, $\mathbf{U}_K^t = \mathbf{W}_V^t \nabla\mathcal{L} \left(\mathbf{W}_V^t\right)^\top \mathbf{W}_Q^t$
    $\mathbf{H}_Q^t = \left(\mathbf{U}_Q^t\right)^\top \mathbf{U}_Q^t + \epsilon\mathbf{I}$, $\mathbf{H}_K^t = \left(\mathbf{U}_K^t\right)^\top \mathbf{U}_K^t + \epsilon\mathbf{I}$
    **Step II. Update** $\mathbf{W}_Q.\mathbf{W}_K$
      $\mathbf{W}_Q^{t+1} = \mathbf{W}_Q^t - \eta\nabla\mathcal{L}(\mathbf{W}_Q^t) \left(\mathbf{H}_Q^t\right)^{-\gamma}$, $\mathbf{W}_K^{t+1} = \mathbf{W}_K^t - \eta\nabla\mathcal{L}(\mathbf{W}_K^t) \left(\mathbf{H}_K^t\right)^{-\gamma}$
  **else**
    **Step III. Update other parameters by** $\mathcal{A}$
      $\mathbf{W}_V^{t+1} = \mathcal{A}\left(\mathbf{W}_V^t, \nabla\mathcal{L}(\mathbf{W}_V^t), \eta_{\text{others}}\right)$
      $\mathbf{W}_{\text{others}}^{t+1} = \mathcal{A}\left(\mathbf{W}_{\text{others}}^t, \nabla\mathcal{L}(\mathbf{W}_{\text{others}}^t), \eta_{\text{others}}\right)$
  **end if**
  $t \leftarrow t + 1$
**end while**

on $\alpha$, resulting in a complementary trade-off between the softmax-gradient approximation and the activation-free approximation. We provide a more explicit discussion of this trade-off in Appendix C. Combining the two stages yields an activation-free approximation of the Attention-specific structural curvature using only available parameters and gradients.

### 3.3 Practical Structure-Aware Preconditioning

In this section, we address the following question: how can the activation-free structural surrogate derived above be converted into a practical optimizer for the Attention parameters? Section 3.1 identifies a bilateral curvature structure for the Query parameter, consisting of the input-dependent factor $\mathbb{E}[\mathbf{X}^\top\mathbf{X}]$ and the Attention-specific factor $\mathbb{E}[\mathbf{K}^\top\mathbf{G}^\top\mathbf{G}\mathbf{K}]$. Our construction focuses on the latter, since it explicitly captures the Query-Key coupling and, as shown in Section 3.2, admits an activation-free approximation using only model parameters and parameter gradients. We therefore use this structural factor as a right-sided preconditioner, avoiding the additional activation-dependent covariance required by the left factor. This yields a practical structure-aware update that directly reflects the Attention coupling identified above.

For the Query projection, the activation-free surrogate derived in Section 3.2 leads to

$$\mathbf{U}_Q = \mathbf{W}_V \nabla\mathcal{L}(\mathbf{W}_V)^\top \mathbf{W}_K, \qquad \mathbf{H}_Q = \mathbf{U}_Q^\top \mathbf{U}_Q + \epsilon\mathbf{I}, \tag{10}$$

where $\epsilon > 0$ is a damping parameter. We then update the Query parameter as

$$\mathbf{W}_Q^{t+1} = \mathbf{W}_Q^t - \eta\nabla\mathcal{L}(\mathbf{W}_Q^t) \left(\mathbf{H}_Q^t\right)^{-\gamma}, \tag{11}$$

where $\eta$ is the learning rate and $\gamma \in (0, 1]$ is the preconditioning order. The Key update is obtained analogously by exchanging the roles of $\mathbf{W}_Q$ and $\mathbf{W}_K$. The complete STRATA procedure is summarized in Algorithm 1. STRATA applies the Attention-specific updates above to the Query and Key projections, while the remaining model parameters are updated by a base optimizer $\mathcal{A}$. The complete procedure is summarized in Algorithm 1.

## 4 Experiments

We empirically evaluate STRATA on LLaMA-style models (Touvron et al., 2023) (130M to 1.3B) to answer two key questions: (1) Can STRATA accelerate pre-training over state-of-the-art matrix optimizers? (2) Does it scale robustly to 1B+ parameters? We benchmark against AdamW and Muon (Liu et al., 2025a). Following (Wen et al., 2025), we carefully tune the hyperparameters of

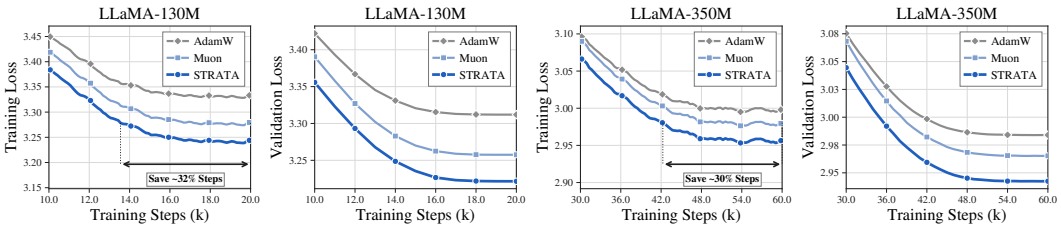



Figure 1: Training and validation curves for LLaMA-130M and LLaMA-350M pre-training. STRATA uses Muon as the base optimizer $\mathcal{A}$ and is compared with vanilla AdamW and Muon.

these baselines to ensure a rigorous comparison. Comprehensive experimental details and setups are deferred to Appendix G.

### 4.1 Acceleration on Standard Transformers: 130M and 350M Scale

We start from the pre-training on LLaMA-130M and LLaMA-350M to verify the acceleration capability of STRATA in the pre-training on models of medium-sized scale. By selecting $\mathcal{A} =$ Muon and comparing STRATA with vanilla AdamW and Muon, we can validate whether STRATA explores an update to $\mathbf{W}_Q$ and $\mathbf{W}_K$ in the standard multi-head Attention structure with better-conditions. Specifically, we train LLaMA-130M and LLaMA-350M on C4/en dataset (Raffel et al., 2020), for 20k and 60k steps (10% warm up) with 5.2B and 15.6B tokens, respectively, with a sequence length of 1024. We set $\gamma = 1/2$, and conduct an ablation experiment to verify the effectiveness of our choice in Appendix G.

We propose the results in Figure 1. The results demonstrate that STRATA consistently and significantly outperforms both the AdamW baseline and the state-of-the-art matrix optimizer, Muon. Notably, AdamW and Muon's convergence noticeably slows while STRATA maintains an efficient descent trajectory due to its precise curvature perception of the Attention coupling structure. This structural preconditioning translates into massive step efficiency gains. To achieve Muon's final training loss value, STRATA saves approximately 6,400 steps, or 32% of the training steps, on the 130M model and 18,000 steps, or 30%, on the 350M model. Furthermore, STRATA's consistently lowest validation loss confirms that this accelerated convergence does not stem from overfitting, but its "gray box" preconditioner effectively guides the model toward flatter local minima with superior generalization capabilities, empirically validating that explicitly incorporating architectural priors decisively overcomes the generic gradient orthogonalization.

### 4.2 Application on 1.3B-Scale Pre-training via Normalized STRATA

We extended STRATA to pre-training tasks for models exceeding 1B parameters and observed the specific characteristics of the algorithm at this scale. Our observations indicated that when the model scale reaches 1B or higher, the increased feature dimensionality causes the spectral radius of the update matrix derived by STRATA to enlarge. To maintain the stability of update magnitudes while preserving the diversity of update directions, we implemented an anisotropic normalization strategy. This strategy is designed to control the spectral radius of the update while avoiding a proportional scaling of eigenvalues across all directions. Accordingly, rather than utilizing a global matrix norm, like the Frobenius norm or 2-norm for normalization, we perform normalization independently for each column of the update matrix. Specifically, we replace the update in Algorithm 1 from $\mathbf{Q}^t = \nabla\mathcal{L}(\mathbf{W}^t)\,(\mathbf{H}^t)^{-\gamma}$ to $\mathcal{N}_{\text{col}}\,(\mathbf{Q}^t) = \left[\frac{\mathbf{q}_1^t}{\|\mathbf{q}_1^t\|_2}, \ldots, \frac{\mathbf{q}_d^t}{\|\mathbf{q}_d^t\|_2}\right]$, where $\mathbf{Q}^t = [\mathbf{q}_1^t, \cdots, \mathbf{q}_d^t]$. This specific column-wise approach has previously been shown to be effective (Glentis et al., 2025).

We evaluate N-STRATA ( $\mathcal{A} =$ Muon) on LLaMA-1.3B using the C4/en dataset ($\sim$40B tokens). As shown in Figure 2, N-STRATA accelerates late-stage convergence compared to AdamW and Muon. Aligning with our theory, while Muon's constant-magnitude updates inevitably induce late-stage stagnation, N-STRATA's decaying approximation error and implicit regularization effectively prevent this plateau. Further stabilized by column-wise normalization curbing extreme updates, N-STRATA sustains its optimization superiority, saving 12,000 training steps over the Muon.

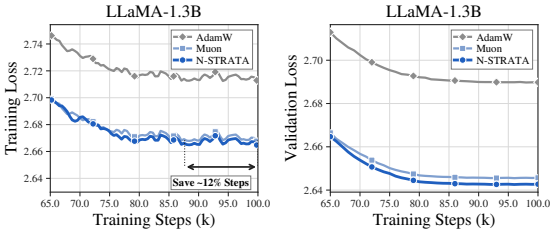



Figure 2: Training and validation loss v.s. steps. Comparison between N-STRATA (with $\mathcal{A} =$ Muon) v.s. AdamW and Muon on LLaMA-1.3B.

**Algorithm 2** Coupled Newton-Schulz Iteration

**Input:** a positive symmetric matrix $\mathbf{A}$, steps $T_1$, $\mathbf{X}^0 = \mathbf{A}/\|\mathbf{A}\|_F$, $\mathbf{Y}^0 = \mathbf{I}$
**while** $t < T_1$ **do**
  $\mathbf{Z}^t = \frac{1}{2}(3\mathbf{I} - \mathbf{Y}^t\mathbf{X}^t)$
  $\mathbf{X}^{t+1} = \mathbf{X}^t\mathbf{Z}^t$, $\mathbf{Y}^{t+1} = \mathbf{Z}^t\mathbf{Y}^t$
  $t \leftarrow t + 1$
**end while**
**Return** $\mathbf{Y}^{T_1}/\sqrt{\|\mathbf{A}\|_F}$

Table 1: Efficiency, including computational overhead, wall-clock time speedup, and memory overhead across different model scales. Base Opt means the optimizer we compare with. Base Steps means the total training steps with the base optimizer. STRATA steps mean the steps (N)-STRATA needs to achieve the same loss level as the base optimizer.

| Model Scale | Base Opt | Base Steps | STRATA Steps | Base t/step (s) | STRATA t/step (s) | Wall-clock Speedup (%) | Base Peak Memory (GB) | STRATA Peak Memory (GB) |
|---|---|---|---|---|---|---|---|---|
| 130M | Muon | 20,000 | 13,600 | 0.74 | 0.75 | 31 | 32.57 | 32.57 |
| 350M | Muon | 60,000 | 42,000 | 1.07 | 1.12 | 23 | 36.53 | 36.52 |
| 1.3B | Muon | 100,000 | 88,000 | 3.27 | 3.40 | 11 | 156.96 | 156.96 |

### 4.3 Computational Overhead and Wall-Clock Time Speedup

While matrix preconditioning often suffers from prohibitive per-step latency due to exact SVD (e.g., Shampoo), STRATA overcomes this bottleneck by employing a coupled Newton-Schulz iteration (see Algorithm 2) to efficiently approximate the $-1/2$ power (Higham, 2008). Inspired by Muon's efficiency paradigm, this approach guarantees a tangible wall-clock speedup, which is especially highly efficient given the small head dimension $d$ (e.g., 64 in LLaMA-350M) of the Attention parameter $\mathbf{W} \in \mathbb{R}^{D\times d}$.

**Proposition 4.1.** *Given a positive symmetric matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$, Algorithm 2 has a quadratic convergence speed to an approximation of $\mathbf{A}^{-1/2}$.*

The proof is shown in Appendix H. In our experiments, we set $T_1 = 5$ and use Algorithm 2 to calculate the $-1/2$ power of the preconditioner. This efficient approximation ensures a substantial, end-to-end wall-clock time speedup throughout the entire training process. In Table 1, we propose the per-step time cost of STRATA and Muon. As shown in Table 1, STRATA consistently achieves significant step efficiency across all model scales. Compared to the Muon baseline, STRATA reduces the required training steps to reach the comparable loss by 6,400, 18,000, and 12,000 steps for the 130M, 350M, and 1.3B models, respectively. While our efficient NS iterations introduce a marginal increase in per-step latency, this slight computational overhead is overwhelmingly offset by the substantial reduction in total convergence steps. Besides, profiling results indicate that STRATA introduces virtually zero additional peak memory overhead. Ultimately, this structural efficiency translates into a tangible end-to-end wall-clock time speedup ranging from 11% to 31%, firmly demonstrating STRATA's practical viability for large-scale model pre-training.

## 5 Conclusion

In this paper, we propose STRATA, a structure-aware preconditioning optimizer that shifts Transformer training from a "black box" to a "gray box" paradigm by explicitly exploiting Query-Key coupling. To ensure application practicality, we design an activation-free approximation and a coupled NS iteration, with small memory cost and latency bottlenecks. Extensive experiments on LLaMA architectures up to 1.3B parameters demonstrate that STRATA significantly accelerates convergence and outperforms state-of-the-art optimizers like AdamW and Muon. This work establishes a rigorous foundation for structure-aware preconditioning, paving the way for future optimizations.

## AI USE STATEMENT

In this work, we used generative AI tools to improve the clarity, grammar, and presentation of the manuscript. All AI-assisted content was carefully reviewed and verified by the authors. All scientific ideas, methodological designs, formulation, theoretical analysis, experimental results, and conclusions were developed and validated by the authors. We take full responsibility for the final content of this work.

## REFERENCES


MS Bartlett. Approximate confidence intervals. *Biometrika*, 40(1/2):12–19, 1953.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311, 2018.

Zachary Charles and Dimitris Papailiopoulos. Stability and generalization of learning algorithms that converge to global optima. In *International conference on machine learning*, pp. 745–754. PMLR, 2018.

Pei Chen. Hessian matrix vs. gauss–newton hessian matrix. *SIAM Journal on Numerical Analysis*, 49(4):1417–1435, 2011.

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

John E Dennis Jr and Robert B Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM, 1996.

Zhehang Du and Weijie Su. The newton-muon optimizer. *arXiv preprint arXiv:2604.01472*, 2026.

Matilde Gargiani, Andrea Zanelli, Moritz Diehl, and Frank Hutter. On the promise of the stochastic generalized gauss-newton method for training dnns. *arXiv preprint arXiv:2006.02409*, 2020.

Athanasios Glentis, Jiaxiang Li, Andi Han, and Mingyi Hong. A minimalist optimizer design for llm pretraining. *arXiv preprint arXiv:2506.16659*, 2025.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pp. 1842–1850. PMLR, 2018.

Moritz Hardt and Tengyu Ma. Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*, 2016.

Nicholas J Higham. *Functions of matrices: theory and computation*. SIAM, 2008.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

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

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Tim Tsz-Kit Lau, Qi Long, and Weijie Su. Polargrad: A class of matrix-gradient optimizers from a unifying preconditioning perspective. *arXiv preprint arXiv:2505.21799*, 2025.

Bingrui Li, Wei Huang, Andi Han, Zhanpeng Zhou, Taiji Suzuki, Jun Zhu, and Jianfei Chen. On the optimization and generalization of two-layer transformers with sign gradient descent. *arXiv preprint arXiv:2410.04870*, 2024.

Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. *Advances in neural information processing systems*, 30, 2017.

Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. *arXiv preprint arXiv:2305.14342*, 2023.

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

Liming Liu, Zhenghao Xu, Zixuan Zhang, Hao Kang, Zichong Li, Chen Liang, Weizhu Chen, and Tuo Zhao. Cosmos: A hybrid adaptive optimizer for memory-efficient training of llms. *arXiv preprint arXiv:2502.17410*, 2025b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pp. 2408–2417. PMLR, 2015.

Arseny Moskvichev, Victor Vikram Odouard, and Melanie Mitchell. The conceptarc benchmark: Evaluating understanding and generalization in the arc domain. *arXiv preprint arXiv:2305.07141*, 2023.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, et al. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the international conference for high performance computing, networking, storage and analysis*, pp. 1–15, 2021.

Mohammad Taher Pilehvar and Jose Camacho-Collados. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1267–1273, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL `http://jmlr.org/papers/v21/20-074.html`.

Adepu Ravi Sankar, Yash Khasbage, Rahul Vigneswaran, and Vineeth N Balasubramanian. A deeper look at the hessian eigenspectrum of deep neural networks and its applications to regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9481–9488, 2021.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Chongjie Si, Debing Zhang, and Wei Shen. Adamuon: Adaptive muon optimizer. *arXiv preprint arXiv:2507.11005*, 2025.

Yuandong Tian, Yiping Wang, Beidi Chen, and Simon S Du. Scan and snap: Understanding training dynamics and token composition in 1-layer transformer. *Advances in neural information processing systems*, 36:71911–71947, 2023.

Tian Tong, Cong Ma, and Yuejie Chi. Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent. *Journal of Machine Learning Research*, 22(150):1–63, 2021.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Mujin Kwun, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. Soap: Improving and stabilizing shampoo using adam. *arXiv preprint arXiv:2409.11321*, 2024.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

Johannes Welbl, Nelson F Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 94–106, 2017.

Kaiyue Wen, David Hall, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them. *arXiv preprint arXiv:2509.02046*, 2025.

Bo Xie, Yingyu Liang, and Le Song. Diverse neural network learns true target functions. In *Artificial Intelligence and Statistics*, pp. 1216–1224. PMLR, 2017.

Xingyu Xu, Yandi Shen, Yuejie Chi, and Cong Ma. The power of preconditioning in overparameterized low-rank matrix sensing. In *International Conference on Machine Learning*, pp. 38611–38654. PMLR, 2023.

Zhuoning Yuan, Yan Yan, Rong Jin, and Tianbao Yang. Stagewise training accelerates convergence of testing error over sgd. *Advances in Neural Information Processing Systems*, 32, 2019.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*, 2018.

Pan Zhou, Hanshu Yan, Xiaotong Yuan, Jiashi Feng, and Shuicheng Yan. Towards understanding why lookahead generalizes better than sgd and beyond. *Advances in Neural Information Processing Systems*, 34:27290–27304, 2021.

# A Detailed Discussions of Representative Matrix Preconditioners

The matrix-form preconditioning view introduced in Section 2 can be understood through several representative optimizers. We begin with K-FAC (Martens & Grosse, 2015), which provides a direct example of how a large parameter-space curvature matrix can be approximated by low-dimensional Kronecker factors. Consider a matrix parameter $\mathbf{W} \in \mathbb{R}^{D\times d}$ acting on an input activation $\mathbf{a} \in \mathbb{R}^D$ through $\mathbf{s} = \mathbf{W}^\top \mathbf{a}$. Let $\mathbf{g} \in \mathbb{R}^d$ denote the derivative of the loss with respect to $\mathbf{s}$. Since $\nabla\mathcal{L}(\mathbf{W}) = \mathbf{a}\mathbf{g}^\top$, the vectorized parameter gradient satisfies $\mathrm{vec}(\nabla\mathcal{L}(\mathbf{W})) = \mathbf{g} \otimes \mathbf{a}$. Therefore, the Fisher matrix associated with this parameter block can be written as

$$\mathbf{F}_{\mathbf{W}} = \mathbb{E}\left[\mathrm{vec}(\nabla\mathcal{L}(\mathbf{W}))\,\mathrm{vec}(\nabla\mathcal{L}(\mathbf{W}))^\top\right] = \mathbb{E}\left[\mathbf{g}\mathbf{g}^\top \otimes \mathbf{a}\mathbf{a}^\top\right]. \tag{12}$$

Under the standard K-FAC approximation that decouples the activation and backpropagated-gradient statistics, this becomes

$$\mathbf{F}_{\mathbf{W}} \approx \mathbb{E}[\mathbf{g}\mathbf{g}^\top] \otimes \mathbb{E}[\mathbf{a}\mathbf{a}^\top]. \tag{13}$$

Thus, instead of manipulating a full curvature matrix in $\mathbb{R}^{Dd\times Dd}$, K-FAC represents the corresponding parameter-block curvature through two substantially smaller matrices. This directly leads to a matrix-form bilateral preconditioner through the standard Kronecker vectorization identity.

Shampoo (Gupta et al., 2018) reaches a similar left-right matrix preconditioning form through a different construction. Rather than explicitly using activation and backpropagated-gradient covariances, Shampoo constructs its factors directly from the gradients of the matrix parameter. Its theoretical justification follows from an upper bound on the accumulated second-moment matrix of the vectorized gradients.

**Proposition A.1** (Lemma 8 of Gupta et al. (2018))**.** *Assume that, for $T$ steps, the gradients of a matrix parameter $\mathbf{W}$ are $\mathbf{G}^1, \ldots, \mathbf{G}^T \in \mathbb{R}^{D\times d}$, each with rank at most $r$. Let $\mathbf{g}^t = \mathrm{vec}(\mathbf{G}^t)$. Then, for any $\epsilon \geq 0$,*

$$\epsilon\mathbf{I}_{Dd} + \frac{1}{r}\sum_{t=1}^{T}\mathbf{g}^t(\mathbf{g}^t)^\top \preceq \underbrace{\left(\epsilon\mathbf{I}_D + \sum_{t=1}^{T}\mathbf{G}^t(\mathbf{G}^t)^\top\right)^{1/2}}_{\mathbf{L}^T} \otimes \underbrace{\left(\epsilon\mathbf{I}_d + \sum_{t=1}^{T}(\mathbf{G}^t)^\top\mathbf{G}^t\right)^{1/2}}_{\mathbf{R}^T}. \tag{14}$$

Accordingly, Shampoo uses $\mathbf{L}^t$ and $\mathbf{R}^t$ as left and right matrix preconditioners for $\mathbf{G}^t$. With the square-root inverse associated with the corresponding second-moment approximation, its matrix-form update takes the form $(\mathbf{L}^t)^{-1/2}\,\mathbf{G}^t\,(\mathbf{R}^t)^{-1/2}$. Therefore, although K-FAC and Shampoo obtain their factors from different quantities, both illustrate how a structured approximation in vectorized parameter space can induce a computationally tractable bilateral matrix preconditioner.

Muon provides another closely related matrix-form update, but its construction is different from the explicit Kronecker factorizations above. Given a matrix gradient $\mathbf{G}$, the Muon update is based on the matrix-sign transformation $\mathbf{G}(\mathbf{G}^\top\mathbf{G})^{-1/2}$, which orthogonalizes the singular directions of the gradient. This update can be computed efficiently through Newton-Schulz iterations, avoiding an exact singular value decomposition. In addition to this matrix-orthogonalization interpretation, recent work (Du & Su, 2026) provides a local second-order interpretation under a structured quadratic approximation.

**Proposition A.2** ((Du & Su, 2026))**.** *Let $\mathbf{W} \in \mathbb{R}^{D\times d}$ be a weight matrix, $\mathbf{Q}$ be an update direction, and $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_R] \in \mathbb{R}^{D\times R}$ denote the sequence-level input activation matrix. Suppose that $\mathcal{L}(\mathbf{W}) = \frac{1}{R}\sum_{i=1}^{R}\mathcal{L}_i(\mathbf{W}^\top\mathbf{a}_i)$. Assuming negligible cross-token curvature and that the feature-dimension curvature of different tokens can be approximated by a shared matrix*

$$\frac{1}{2}\mathbf{H} \approx \int_0^1 (1-t)\nabla^2\mathcal{L}_i\left(\mathbf{W}^\top\mathbf{a}_i - t\mathbf{Q}^\top\mathbf{a}_i\right)\mathrm{d}t, \tag{15}$$

*the parameter-space Hessian $\mathcal{H}_{\mathbf{W}} := \nabla^2_{\mathrm{vec}(\mathbf{W})}\mathcal{L}(\mathbf{W})$ admits the approximation*

$$\mathcal{H}_{\mathbf{W}} \approx \frac{1}{R}\left(\mathbf{A}\mathbf{A}^\top\right) \otimes \mathbf{H}. \tag{16}$$

The proof is provided in Appendix I. Proposition A.2 gives one possible second-order interpretation of matrix orthogonalization under the stated local approximation. In particular, the resulting quadratic model relates the update geometry to both the parameter gradient and the input covariance. Under an additional activation-whitening approximation, this dependence on the input covariance simplifies, recovering the matrix-sign form used by Muon.

Taken together, these examples clarify the distinction emphasized in Section 2. Kronecker structure provides a principled route from an intractable parameter-space curvature matrix to tractable left and right matrix factors, but different optimizers obtain these factors from different sources. K-FAC constructs them from activation and backpropagated-gradient statistics, Shampoo from accumulated parameter-gradient statistics, while Muon applies matrix orthogonalization and admits a related local second-order interpretation under additional assumptions. This distinction between the *representation* of matrix-form preconditioning and the *source* of the information used to construct it motivates our subsequent focus on architecture-specific factors derived directly from the Attention mechanism.

# B Notations and Standard Assumptions

We use bold lowercase letters, such as $\mathbf{x}$, to denote vectors and bold uppercase letters, such as $\mathbf{X}$, to denote matrices. Scalars and constants are written in non-bold symbols. For any matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, $[\mathbf{X}]_{ij}$ denotes its $(i,j)$-th entry, and for any vector $\mathbf{x} \in \mathbb{R}^d$, $x_i$ denotes its $i$-th entry. The transpose of a vector or matrix is denoted by $^\top$.

For any square matrix $\mathbf{X} \in \mathbb{R}^{d\times d}$, $\mathrm{tr}(\mathbf{X})$ denotes its trace. For any vector $\mathbf{x} \in \mathbb{R}^d$, $\mathrm{diag}(\mathbf{x}) \in \mathbb{R}^{d\times d}$ denotes the diagonal matrix whose diagonal entries are given by $\mathbf{x}$. For any matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{m\times n}$, their Frobenius inner product is defined as $\langle \mathbf{X}, \mathbf{Y}\rangle_{\mathrm{F}} := \mathrm{tr}(\mathbf{X}^\top \mathbf{Y})$, and the corresponding Frobenius norm is denoted by $\|\mathbf{X}\|_{\mathrm{F}}$.

We use $\mathrm{vec}(\mathbf{X})$ to denote the vectorization of a matrix $\mathbf{X}$. For two matrices $\mathbf{X}$ and $\mathbf{Y}$, $\mathbf{X} \otimes \mathbf{Y}$ denotes their Kronecker product, and $\mathbf{X} \odot \mathbf{Y}$ denotes their Hadamard product.

**Proposition B.1** ($L$-Lipschitz smooth)**.** *Let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ be $L$-Lipschitz smooth, i.e., there exists a constant $L > 0$ such that*

$$\forall \mathbf{X}, \mathbf{Y} \in \mathbb{R}^{m\times n}, \|\nabla f(\mathbf{X}) - \nabla f(\mathbf{Y})\|_F \le L\|\mathbf{X} - \mathbf{Y}\|_F.$$

**Proposition B.2** ($\mu$-strong convexity)**.** *Let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ be $\mu$-Strong Convex, i.e., there exists a constant $\mu > 0$ such that $\forall \mathbf{X}, \mathbf{Y} \in \mathbb{R}^{m\times n}$,*

$$\langle \nabla f(\mathbf{X}) - \nabla f(\mathbf{Y}), \mathbf{X} - \mathbf{Y}\rangle_F \ge \mu\|\mathbf{X} - \mathbf{Y}\|_F^2.$$

Note that $\mu$-strong convexity implies the $\mu$-Polyak–Łojasiewicz (PŁ) condition or inequality:

$$\forall \mathbf{X} \in \mathbb{R}^{m\times n}, \|\nabla f(\mathbf{X})\|_F^2 \ge 2\mu\left(f(\mathbf{X}) - f^*\right),$$

where $f^* = \min f$, which is a more relaxed condition compared with $\mu$-strong convexity.

# C Detailed Discussions on Approximation and Optimization

## C.1 Complementary Trade-off of the Two Approximation Stages

Theorems 3.3 and 3.4 characterize the two approximation steps introduced in Section 3.2. The first result controls the discrepancy introduced by replacing the exact softmax-logit gradient $\mathbf{G}$ with its main term $\mathbf{M}$, whereas the second result controls the discrepancy between the activation-dependent target $\mathcal{K}_{\mathrm{target}}$ and the activation-free surrogate $\mathcal{K}_{\mathrm{approx}}$. Importantly, the two bounds exhibit opposite dependence on the softmax concentration parameter $\alpha$.

To make this dependence explicit, suppose $\|\mathbf{M}\|_F \le D_1$ and denote $r_1(\alpha) = D_1 R^{\frac{1-\alpha}{2}}$. Theorem 3.3 gives an approximation bound proportional to $r_1(\alpha)$, which decreases monotonically with $\alpha$. Therefore, under the softmax distribution model considered in the theorem, the approximation $\mathbf{G} \approx \mathbf{M}$ becomes tighter as $\alpha$ increases.

For the second approximation, let $C_0 = \delta^2 D_0 D^2$ and denote $r_2(\alpha) = C_0 R^{\frac{1}{2}} (1 - R^{-\alpha})$. Theorem 3.4 gives a bound proportional to $r_2(\alpha)$, which increases monotonically with $\alpha$. Consequently, the activation-free approximation becomes tighter as $\alpha$ decreases.

Hence, the two stages exhibit complementary dependence on the same concentration parameter: larger $\alpha$ favors the softmax-gradient approximation in Stage I, whereas smaller $\alpha$ favors the activation-free approximation in Stage II. This observation provides a more detailed view of the trade-off summarized in Section 3.2. We emphasize that the two theorems control different approximation quantities and involve different scale factors. Therefore, their opposite monotonic dependence does not by itself imply a uniform bound on the combined approximation error. Rather, it characterizes how the two sources of approximation error respond differently to the concentration of the Attention distribution.

### C.2 General Convergence Implications

Since exact convergence analysis for full Transformers remains an open problem, we characterize STRATA under the standard framework of preconditioned gradient descent (Lau et al., 2025). Assuming Lipschitz smoothness and the Polyak-Łojasiewicz condition, we consider the symmetric update of $\mathbf{W}$ using the generalized preconditioner form $\mathbf{H} = \mathbf{U}^\top \mathbf{U} + \epsilon \mathbf{I}$. Our result characterizes how the spectrum of $\mathbf{H}$ interacts with the objective condition number $\kappa$ in the resulting preconditioned update. We first recall the standard results for functions satisfying $L$-Lipschitz smoothness, $\mu$-strong convexity, and the $\mu$-PL condition in Appendix B. Then we make the following assumption for our objective.

**Assumption C.1.** We assume that the objective function $\mathcal{L} : \mathbb{R}^{D\times d} \to \mathbb{R}$ is $L$-Lipschitz smooth and a $\mu$-PL function. Without loss of generality, we let $d \le D$. Let $\mathcal{L}^* = \min \mathcal{L}$, and we define the symmetric positive-definition preconditioner for update as $\mathbf{H}^t = (\mathbf{U}^t)^\top \mathbf{U}^t + \epsilon \mathbf{I} \in \mathbb{R}^{d\times d}$ with the order $-\gamma, \gamma \in (0, 1]$. Thus the update can be represented as

$$\mathbf{W}^{t+1} = \mathbf{W}^t - \eta \nabla \mathcal{L}(\mathbf{W}^t) \left(\mathbf{H}^t\right)^{-\gamma}. \tag{17}$$

We also denote $\{\sigma_i^t : i \in [d]\}$ as the singular values of $\mathbf{U}^t$, and $\kappa = L/\mu$ be the condition number.

Under Assumption C.1, we propose our main convergence analysis.

**Theorem C.2.** *Suppose Assumption C.1 holds. For the fixed learning rate* $\eta = \frac{\min_{t=0}^{T} \min_{i=1}^{d} (\lambda_i^t)^\gamma}{L}$, *the iterates of STRATA* (17) *satisfies*

$$\mathcal{L}\left(\mathbf{W}^T\right) - \mathcal{L}^* \le \left(1 - \frac{1}{\tilde{\kappa}_\gamma \kappa}\right)^T \left(\mathcal{L}(\mathbf{W}^0) - \mathcal{L}^*\right) \tag{18}$$

*for* $T$ *iterations with* $\tilde{\kappa}_\gamma = \frac{\max_{t=0}^{T} \max_{i=1}^{d} (\lambda_i^t)^\gamma}{\min_{t=0}^{T} \min_{i=1}^{d} (\lambda_i^t)^\gamma}$, $\lambda_i^t = (\sigma_i^t)^2 + \epsilon$ *for all* $i \in [d]$ *and* $t \in [T]$.

The proof is shown in Appendix F. Theorem C.2 provides a general convergence characterization for right-preconditioned gradient descent under the stated smoothness and Polyak–Łojasiewicz assumptions. In particular, the convergence rate is governed jointly by the objective condition number $\kappa$ and the spectral spread $\tilde{\kappa}_\gamma$ of the preconditioner along the optimization trajectory. Therefore, the result isolates the conditioning of the preconditioner as a key quantity controlling the behavior of this class of updates.

It is informative to contrast this dependence with that of matrix-sign descent. Recent analysis of Muon-style updates (Lau et al., 2025) considers $\mathbf{Q}^t = \mathrm{msign}(\nabla \mathcal{L}(\mathbf{W}^t))$. Unlike the right-preconditioned gradient in (17), the matrix-sign transformation normalizes the singular values of the gradient and therefore does not preserve its original magnitude. In particular, $\|\mathbf{Q}^t\|_F^2 = \mathrm{rank}(\nabla \mathcal{L}(\mathbf{W}^t))$. Let $r_{\max} = \max_{t=0}^{T} \mathrm{rank}(\nabla \mathcal{L}(\mathbf{W}^t))$. Under the same smoothness and Polyak–Łojasiewicz conditions, the corresponding optimality gap $\Delta^t := \mathcal{L}(\mathbf{W}^t) - \mathcal{L}^*$ admits the recursion

$$\Delta^{t+1} \le \Delta^t - \eta \sqrt{2\mu \Delta^t} + \frac{L}{2} \eta^2 r_{\max}. \tag{19}$$

For a fixed learning rate, the last term does not vanish as $\Delta^t$ decreases, which motivates the use of learning-rate adaptation in analyses of matrix-sign descent.

Moreover, recent literature (Lau et al., 2025) shows that, with an appropriately designed adaptive learning-rate schedule, Muon can achieve a linear convergence rate whose dependence involves both the rank and the condition number of the gradient matrix. By comparison, Theorem C.2 characterizes the right-preconditioned update through $\tilde{\kappa}_\gamma$ and the objective condition number $\kappa$. These bounds describe different update geometries and should not be interpreted as a direct theoretical ordering between STRATA and Muon. For STRATA, the relevant question is whether the architecture-specific preconditioner exhibits a favorable spectrum in practice, which we investigate empirically through the spectral diagnostics in Figure 5 (a).

*Remark* C.3. We emphasize that the PL condition is significantly weaker than strong convexity and has emerged as a widely adopted framework for studying non-convex optimization (Yuan et al., 2019; Zhou et al., 2021; Lau et al., 2025). Prior work also considers local or trajectory-dependent PL behavior in analyses of neural network optimization (Hardt & Ma, 2016; Xie et al., 2017; Li & Yuan, 2017; Charles & Papailiopoulos, 2018). Therefore, the result above should be interpreted as a conditional convergence characterization under the stated smoothness and PL assumptions rather than as a global convergence guarantee for full Transformer training.

## D Deferred Proofs in Section 3.1

Before proof of Lemma 3.1, we start our analysis by leveraging the structure of loss to design a biased estimator of the Hessian of softmax logits $\mathrm{vec}(\mathbf{A})$. Suppose $\mathcal{L}(f(\mathbf{Z}), \mathbf{y})$ is a cross-entropy loss on logits $f(\mathbf{Z}) \in \mathbb{R}^C$ where $C$ is the number of classes in a multi-class classification problem (e.g., the vocabulary size in LLMs.) Denoting $\mathbf{S} = \mathrm{softmax}_{\text{row}}\left(\mathbf{X}\mathbf{W}_Q\mathbf{W}_K^\top\mathbf{X}^\top\right)$, the Hessian of $\mathcal{L}$ w.r.t. to softmax logits $\mathrm{vec}(\mathbf{A})$, $\nabla^2_{\mathrm{vec}(\mathbf{A})}\mathcal{L}$, has a Gauss-Newton decomposition:

$$\nabla^2_{\mathrm{vec}(\mathbf{A})}\mathcal{L} = \mathbf{J}_{\mathbf{A}}f^\top \cdot \nabla^2_f\mathcal{L} \cdot \mathbf{J}_{\mathbf{A}}f + \mathbf{J}_{\mathbf{AA}}f \cdot \frac{\partial \mathcal{L}}{\partial f} \tag{20}$$

where $\mathbf{J}_{\mathbf{A}}f = \frac{\partial f}{\partial\, \mathrm{vec}(\mathbf{A})} \in \mathbb{R}^{C\times R^2}$ is the Jacobian of $f$ w.r.t. to softmax logits $\mathrm{vec}(\mathbf{A})$. $\nabla^2_f\mathcal{L} = \frac{\partial^2\mathcal{L}}{\partial f^2}$ is the second-order derivatives of the loss w.r.t to the logits, and $\mathbf{J}_{\mathbf{AA}}f$ is the second-order derivatives of the multi-variate function $f$ w.r.t $\mathrm{vec}(\mathbf{A})$, viewed as a linear map from $\mathbb{R}^{R^2\times R^2}$ to $\mathbb{R}^C$.

In the context of neural networks, past works have found that the second term in (20) is often relative smaller than the first term (Sankar et al., 2021). Thus we often refer the first term as the Gauss-Newton matrix (Dennis Jr & Schnabel, 1996; Chen, 2011) and used as preconditioners in second-order optimizers (Martens, 2020; Gargiani et al., 2020; Liu et al., 2023). Following this line of work, we can rewrite the equation (20) as

$$\nabla^2_{\mathrm{vec}(\mathbf{A})}\mathcal{L} \approx \mathbf{J}_{\mathbf{A}}f^\top \cdot \nabla^2_f\mathcal{L} \cdot \mathbf{J}_{\mathbf{A}}f. \tag{21}$$

Moreover, we claim that considering the cross-entropy loss, $\mathbf{J}_{\mathbf{A}}f^\top \cdot \nabla^2_f\mathcal{L} \cdot \mathbf{J}_{\mathbf{A}}$ can be represented as the expectation of a vector-spanned matrix as below.

**Lemma D.1.** *Suppose $\mathcal{L}$ is the cross-entropy loss. Denote $\mathbf{J}_{\mathbf{A}}f = \frac{\partial f}{\partial\, \mathrm{vec}(\mathbf{A})} \in \mathbb{R}^{C\times R^2}$ is the Jacobian of $f$ w.r.t. to softmax logits $\mathrm{vec}(\mathbf{A})$. $\nabla^2_f\mathcal{L} = \frac{\partial^2\mathcal{L}}{\partial f^2}$ is the second-order derivatives of the loss w.r.t to the logits. Letting $\frac{\partial \mathcal{L}(f(\mathbf{Z}),\mathbf{y})}{\partial\, \mathrm{vec}(\mathbf{A})} = \mathrm{vec}(\mathbf{G})$, we have*

$$\mathbf{J}_{\mathbf{A}}f^\top \cdot \nabla^2_f\mathcal{L} \cdot \mathbf{J}_{\mathbf{A}} = \mathbb{E}_{\mathbf{y}\sim\mathrm{Cat}(f(\mathbf{Z}))}\left[\frac{\partial \mathcal{L}(f(\mathbf{Z}),\mathbf{y})}{\partial\, \mathrm{vec}(\mathbf{A})} \cdot \left(\frac{\partial \mathcal{L}(f(\mathbf{Z}),\mathbf{y})}{\partial\, \mathrm{vec}(\mathbf{A})}\right)^\top\right],$$

*where $\mathrm{Cat}(x)$ represents the categorical distribution with parameter $x$.*

Based on Lemma D.1, we can leverage the Kronecker factorization foundation discussed in Section 2 to decompose the expected Hessian and construct a structure-aware preconditioner. We formalize this transition from the parameter-space Hessian to the matrix-form update in the following Lemma.

*Proof of Lemma D.1.* We notice that $\nabla^2_f\mathcal{L}$ only depends on $f$ but not $\mathbf{y}$. It implies $\nabla^2_f\mathcal{L} = \mathbb{E}_{\mathbf{y}\sim p(\mathbf{Z})}\left[\frac{\partial^2\mathcal{L}(f(\mathbf{Z}),\mathbf{y})}{\partial f^2(\mathbf{Z})}\right]$, where $p(\mathbf{Z}) = \mathrm{softmax}(f(\mathbf{Z}))$. Because $\mathcal{L}$ is the negative log-probability

of the probabilistic model defined by the categorical distribution $\mathrm{Cat}(f(\mathbf{Z}))$,by Bartlett's second identity (Bartlett, 1953),we have

$$\begin{aligned}\nabla_f^2 \mathcal{L} &= \mathbb{E}_{\mathbf{y}\sim\mathrm{Cat}(f(\mathbf{Z}))}\left[\frac{\partial^2 \mathcal{L}(f(\mathbf{Z}),\mathbf{y})}{\partial f^2(\mathbf{Z})}\right] \\ &= \mathbb{E}_{\mathbf{y}\sim\mathrm{Cat}(f(\mathbf{Z}))}\left[\frac{\partial \mathcal{L}(f(\mathbf{Z}),\mathbf{y})}{\partial f(\mathbf{Z})}\cdot\left(\frac{\partial \mathcal{L}(f(\mathbf{Z}),\mathbf{y})}{\partial f(\mathbf{Z})}\right)^\top\right].\end{aligned} \tag{22}$$

Thus we have

$$\begin{aligned}&\mathbf{J}_{\mathbf{A}} f^\top \cdot \nabla_f^2 \mathcal{L} \cdot \mathbf{J}_{\mathbf{A}} \\ &= \mathbb{E}_{\mathbf{y}\sim\mathrm{Cat}(f(\mathbf{Z}))}\left[\mathbf{J}_{\mathbf{A}} f^\top \frac{\partial \mathcal{L}(f(\mathbf{Z}),\mathbf{y})}{\partial f(\mathbf{Z})}\cdot\left(\frac{\partial \mathcal{L}(f(\mathbf{Z}),\mathbf{y})}{\partial f(\mathbf{Z})}\right)^\top \mathbf{J}_{\mathbf{A}} f\right] \\ &= \mathbb{E}_{\mathbf{y}\sim\mathrm{Cat}(f(\mathbf{Z}))}\left[\frac{\partial \mathcal{L}(f(\mathbf{Z}),\mathbf{y})}{\partial \operatorname{vec}(\mathbf{A})}\cdot\left(\frac{\partial \mathcal{L}(f(\mathbf{Z}),\mathbf{y})}{\partial \operatorname{vec}(\mathbf{A})}\right)^\top\right].\end{aligned} \tag{23}$$

□

Now we formally prove Lemma 3.1.

*Proof of Lemma 3.1.* Based on Lemma D.1, and denoting $\frac{\partial \mathcal{L}(f(\mathbf{Z}),y)}{\partial \operatorname{vec}(\mathbf{A})} = \operatorname{vec}(\mathbf{G})$, the expected Gauss-Newton Hessian of $\mathbf{W}_Q$ can be written as:

$$\mathbb{E}[\mathcal{H}_{\operatorname{vec}(\mathbf{W}_Q)}] \approx \mathbb{E}_{(\mathbf{X},y)\sim\mathcal{D}}\left[(\mathbf{K}\otimes\mathbf{X})^\top \operatorname{vec}(\mathbf{G})\operatorname{vec}(\mathbf{G})^\top(\mathbf{K}\otimes\mathbf{X})\right].$$

Using the standard Kronecker product vectorization identity $(\mathbf{B}^\top\otimes\mathbf{A})\operatorname{vec}(\mathbf{C}) = \operatorname{vec}(\mathbf{ACB})$, we have $(\mathbf{K}\otimes\mathbf{X})^\top\operatorname{vec}(\mathbf{G}) = (\mathbf{K}^\top\otimes\mathbf{X}^\top)\operatorname{vec}(\mathbf{G}) = \operatorname{vec}(\mathbf{X}^\top\mathbf{GK})$. Therefore, we obtain

$$\mathbb{E}[\mathbf{H}(\mathbf{W}_Q)] \approx \mathbb{E}_{(\mathbf{X},y)\sim\mathcal{D}}\left[\operatorname{vec}(\mathbf{X}^\top\mathbf{GK})\operatorname{vec}(\mathbf{X}^\top\mathbf{GK})^\top\right].$$

Due to assuming statistical independence between the activations and the gradients, we can decouple the expectation

$$\mathbb{E}[\mathcal{H}_{\operatorname{vec}(\mathbf{W}_Q)}] \approx \mathbb{E}\left[\mathbf{K}^\top\mathbf{G}^\top\mathbf{GK}\right]\otimes\mathbb{E}\left[\mathbf{X}^\top\mathbf{X}\right].$$

Let $\tilde{\mathcal{H}}_{\operatorname{vec}(\mathbf{W}_Q)}$ denote this approximated preconditioner. When we apply its fractional inverse power $-\gamma$ to the vectorized gradient of $\mathbf{W}_Q$, we utilize the mixed-product property of Kronecker products $(\mathbf{A}\otimes\mathbf{B})^{-\gamma} = \mathbf{A}^{-\gamma}\otimes\mathbf{B}^{-\gamma}$ and obtain

$$\left(\tilde{\mathbf{H}}(\mathbf{W}_Q)\right)^{-\gamma} = \left(\mathbb{E}\left[\mathbf{K}^\top\mathbf{G}^\top\mathbf{GK}\right]\right)^{-\gamma}\otimes\left(\mathbb{E}\left[\mathbf{X}^\top\mathbf{X}\right]\right)^{-\gamma}$$

Applying this operator to $\operatorname{vec}(\nabla\mathcal{L}(\mathbf{W}_Q))$ and reusing the vectorization identity $(\mathbf{B}^\top\otimes\mathbf{A})\operatorname{vec}(\mathbf{C}) = \operatorname{vec}(\mathbf{ACB})$, we obtain

$$\left(\tilde{\mathbf{H}}(\mathbf{W}_Q)\right)^{-\gamma}\operatorname{vec}(\nabla\mathcal{L}(\mathbf{W}_Q)) = \operatorname{vec}\left(\left(\mathbb{E}\left[\mathbf{X}^\top\mathbf{X}\right]\right)^{-\gamma}\nabla\mathcal{L}(\mathbf{W}_Q)\left(\mathbb{E}\left[\mathbf{K}^\top\mathbf{G}^\top\mathbf{GK}\right]\right)^{-\gamma}\right).$$

This completes the proof. □

## E Deferred Proofs in Section 3.2

*Proof of Lemma 3.2.* We try to find a matrix $\mathbf{G}\in\mathbb{R}^{R\times R}$ that satisfies

$$\mathrm{d}\mathcal{L} = \operatorname{Tr}\left(\mathbf{G}^\top\mathrm{d}\mathbf{A}\right). \tag{24}$$

We first try to find $\bar{\mathbf{S}}$ such that $\mathrm{d}\mathcal{L} = \operatorname{Tr}\left(\bar{\mathbf{S}}^\top\mathrm{d}\mathbf{S}\right)$. Based on the chain rule, we have

$$\mathrm{d}\mathcal{L} = \operatorname{Tr}\left(\frac{\partial\mathcal{L}}{\partial\mathbf{Z}}^\top\mathrm{d}\mathbf{Z}\right), \tag{25}$$

and

$$\mathrm{d}\mathbf{Z} = \mathrm{d}\mathbf{S} \cdot \mathbf{X}\mathbf{W}_V. \tag{26}$$

Thus we have

$$\mathrm{d}\mathcal{L} = \mathrm{Tr}\left(\frac{\partial \mathcal{L}}{\partial \mathbf{Z}}^\top \mathrm{d}\mathbf{S} \cdot \mathbf{X}\mathbf{W}_V\right) = \mathrm{Tr}\left(\mathbf{X}\mathbf{W}_V \frac{\partial \mathcal{L}}{\partial \mathbf{Z}}^\top \mathrm{d}\mathbf{S}\right), \tag{27}$$

which implies that $\bar{\mathbf{S}} = \frac{\partial \mathcal{L}}{\partial \mathbf{Z}}\mathbf{W}_V^\top \mathbf{X}^\top$. Then we analyze $\mathrm{d}\mathbf{S}$. For $\mathbf{S} = \mathrm{softmax}(\mathbf{A})$, letting $\mathbf{S} = \left[\mathbf{s}_1^\top, \cdots, \mathbf{s}_R^\top\right]^\top$, $\bar{\mathbf{S}} = \left[\bar{\mathbf{s}}_1^\top, \cdots, \bar{\mathbf{s}}_R^\top\right]^\top$, $\mathbf{A} = \left[\mathbf{a}_1^\top, \cdots, \mathbf{a}_R^\top\right]^\top$ and $\mathbf{G} = \left[\mathbf{g}_1^\top, \cdots, \mathbf{g}_R^\top\right]^\top$, for $i$-th row with $i \in [R]$, we have

$$g_{ik} = \sum_{j=1}^{R} \frac{\partial \mathcal{L}}{\partial s_{ij}} \frac{\partial s_{ij}}{\partial a_{ik}}. \tag{28}$$

Based on the derivation of $\mathrm{softmax}$, we have $\frac{\partial s_{ij}}{\partial a_{ik}} = s_{ij}(\delta_{jk} - s_{jk})$, where $\delta_{jk} = 1$ if $j = k$ else $0$. Then we have

$$\begin{aligned} g_{ik} &= \sum_{j=1}^{R} \bar{s}_{ij}[s_{ij}(\delta_{jk} - s_{jk})] \\ &= \bar{s}_{ik}s_{ik} - s_{ik}\sum_{j=1}^{R} \bar{s}_{ij}s_{ij} \\ &= \bar{s}_{ik}s_{ik} - \left(\bar{\mathbf{s}}_i \mathbf{s}_i^\top\right) s_{ik} \end{aligned} \tag{29}$$

It implies that the $i$-th row of $\mathbf{G}$ satisfies

$$\mathbf{g}_i = \bar{\mathbf{s}}_i \odot \mathbf{s}_i - \left(\bar{\mathbf{s}}_i \mathbf{s}_i^\top\right) \mathbf{s}_i. \tag{30}$$

Thus matrix $\mathbf{G}$ satisfies

$$\mathbf{G} = \bar{\mathbf{S}} \odot \mathbf{S} - \mathbf{D}\mathbf{S}, \tag{31}$$

where $\mathbf{D} = \bar{\mathbf{S}}\mathbf{S}^\top \odot \mathbf{I}$ is the diagonal of $\bar{\mathbf{S}}\mathbf{S}^\top$. Applying $\bar{\mathbf{S}} = \frac{\partial \mathcal{L}}{\partial \mathbf{Z}}\mathbf{W}_V^\top \mathbf{X}^\top$ we finish the proof. □

*Proof of Theorem 3.3.* Let $\mathbf{S} = \left[\mathbf{s}_1^\top, \cdots, \mathbf{s}_R^\top\right]^\top$, $\bar{\mathbf{S}} = \left[\bar{\mathbf{s}}_1^\top, \cdots, \bar{\mathbf{s}}_R^\top\right]^\top$. Denote $\mathbf{m}_i$ and $\mathbf{n}_i$ be the $i$-th row of $\mathbf{M}$ and $\mathbf{N}$. We have

$$\|\mathbf{m}_i\|^2 = \sum_{j=1}^{R} \left(\bar{s}_{ij}s_{ij}\right)^2, \tag{32}$$

and

$$\|\mathbf{n}_i\|^2 = \left(\sum_{j=1}^{R} \bar{s}_{ij}s_{ij}\right)^2 \left(\sum_{j=1}^{R} s_{ij}^2\right). \tag{33}$$

Let $u_j = \bar{s}_{ij}s_{ij}$, we have

$$\rho_i^2 = \frac{\|\mathbf{n}_i\|^2}{\|\mathbf{m}_i\|^2} = \frac{\left(\sum_{j=1}^{R} u_j\right)^2}{\sum_{j=1}^{R} u_j^2} \|\mathbf{s}_i\|^2 \leq R\|\mathbf{s}_i\|^2. \tag{34}$$

We notice that

$$\|\mathbf{s}_i\|^2 = kp_1^2 + (R-k)p_0^2, \tag{35}$$

for each $i \in [R]$, which implies

$$\frac{\|\mathbf{N}\|_F^2}{\|\mathbf{M}\|_F^2} = \frac{\sum_{i=1}^{R} \|\mathbf{n}_i\|^2}{\sum_{i=1}^{R} \|\mathbf{m}_i\|^2} \leq R\left(kp_1^2 + (R-k)p_0^2\right). \tag{36}$$

Thus we have

$$\begin{aligned}\frac{\|\mathbf{N}\|_F}{\|\mathbf{M}\|_F} &\leq \sqrt{R}\cdot\left(\sqrt{k}p_1+\sqrt{R-k}p_0\right)\\ &= \sqrt{R}\left(\frac{(1-p_0R)}{\sqrt{k}}+\left(\sqrt{k}+\sqrt{R-k}\right)p_0\right)\\ &\leq \frac{\sqrt{R}(1-p_0R)}{\sqrt{k}}+\sqrt{2}Rp_0.\end{aligned} \tag{37}$$

For the first inequality we use $\sqrt{A+B}\leq\sqrt{A}+\sqrt{B}$ for $A,B\geq 0$. For the third inequality we use $\sqrt{A}+\sqrt{B}\leq\sqrt{2(A+B)}$ for $A,B\geq 0$. Applying $p=\mathcal{O}\left(R^{-\beta}\right)$ and $k=\Theta\left(R^{\alpha}\right)$, we have

$$\|\mathbf{N}\|_F\leq\mathcal{O}\left(R^{\frac{1-\alpha}{2}}\right)\|\mathbf{M}\|_F. \tag{38}$$

Therefore, we have

$$\|\mathbf{M}-\mathbf{G}\|_F\leq\mathcal{O}\left(R^{\frac{1-\alpha}{2}}\right)\|\mathbf{M}\|_F. \tag{39}$$

We finish the proof. □

*Proof of Theorem 3.4.* We consider each element in $\mathcal{K}_{\text{target}}$ and $\mathcal{K}_{\text{approx}}$, respectively. First, denote $\mathbf{Y}=\mathbf{G}\mathbf{X}$, $\mathbf{S}=\left[\mathbf{s}_1^\top,\cdots,\mathbf{s}_R^\top\right]^\top$, $\mathbf{Q}=\left[\mathbf{q}_1^\top,\cdots,\mathbf{q}_R^\top\right]^\top$, $\mathbf{X}=\left[\mathbf{x}_1^\top,\cdots,\mathbf{x}_R^\top\right]^\top$. Letting $[\mathbf{X}]_{ij}$, $[\mathbf{Y}]_{ij}$ represents the element in $i$-th row and $j$-th column of $\mathbf{X}$ and $\mathbf{Y}$, we have

$$[\mathbf{Y}]_{ia}=\sum_{j=1}^{R}\left(\mathbf{q}_i\mathbf{x}_j^\top s_{ij}\right)[\mathbf{X}]_{ja}=\mathbf{q}_i\left(\sum_{j=1}^{R}s_{ij}[\mathbf{X}]_{ja}\mathbf{x}_j^\top\right). \tag{40}$$

Letting $\mathbf{v}_{ia}^\top=\sum_{j=1}^{R}s_{ij}[\mathbf{X}]_{ja}\mathbf{x}_j^\top$, we have

$$\begin{aligned}[\mathcal{K}_{\text{target}}]_{ab}&=\sum_{i=1}^{R}[\mathbf{Y}]_{ia}[\mathbf{Y}]_{ib}\\ &=\sum_{i=1}^{R}\left(\mathbf{q}_i\mathbf{v}_{ia}^\top\right)\left(\mathbf{q}_i\mathbf{v}_{ib}^\top\right)\\ &=\sum_{i=1}^{R}\sum_{j,l=1}^{R}s_{ij}s_{il}[\mathbf{X}]_{ja}[\mathbf{X}]_{lb}\left(\mathbf{q}_i\mathbf{x}_j^\top\right)\left(\mathbf{q}_i\mathbf{x}_l^\top\right).\end{aligned} \tag{41}$$

Similarly, denoting $\mathbf{U}=\mathbf{Q}^\top\mathbf{S}\mathbf{X}$, we have

$$[\mathbf{U}]_{da}=\sum_{i=1}^{R}\left[[\mathbf{Q}]_{id}\left(\sum_{j=1}^{R}s_{ij}[\mathbf{X}]_{ja}\right)\right]. \tag{42}$$

Letting $c_{ia}=\sum_{j=1}^{R}s_{ij}[\mathbf{X}]_{ja}$, because $\mathcal{K}_{\text{approx}}=\mathbf{U}^\top\mathbf{U}$, we have

$$\begin{aligned}[\mathcal{K}_{\text{approx}}]_{ab}&=\left(\sum_{i=1}^{R}c_{ia}\mathbf{q}_i^\top\right)\left(\sum_{m=1}^{R}c_{mb}\mathbf{q}_m\right)\\ &=\sum_{i,m=1}^{R}\sum_{j,l=1}^{R}s_{ij}s_{ml}[\mathbf{X}]_{ja}[\mathbf{X}]_{lb}\mathbf{q}_i\mathbf{q}_m^\top.\end{aligned} \tag{43}$$

We try to bound $\Delta=\mathcal{K}_{\text{target}}-\mathcal{K}_{\text{approx}}$. Considering $[\Delta]_{ab}$, the gap can be classified into two cases: $i=m$ and $i\neq m$; that is

$$\mathbb{E}\left[[\Delta]_{ab}\right]=\underbrace{\mathbb{E}\left[[\Delta]_{ab}|i=m\right]}_{\mathcal{I}_1}+\underbrace{\mathbb{E}\left[[\Delta]_{ab}|i\neq m\right]}_{\mathcal{I}_2}. \tag{44}$$

Based on assumptions in the Theorem, we have

$$\begin{aligned}\mathcal{I}_1 &= \mathbb{E}\left[\sum_{i=1}^{R}\sum_{j,l=1}^{R} s_{ij}s_{il}[\mathbf{X}]_{ja}[\mathbf{X}]_{lb}\left(\left(\mathbf{q}_i\mathbf{x}_j^\top\right)\left(\mathbf{q}_i\mathbf{x}_l^\top\right) - \mathbf{q}_i\mathbf{q}_i^\top\right)\right] \\ &\leq \delta^2\sum_{i=1}^{R}\sum_{j,l=1}^{R} s_{ij}s_{il}\mathbf{q}_i\mathbb{E}\left[\mathbf{x}_j^\top\mathbf{x}_l - \mathbf{I}\right]\mathbf{q}_i^\top \\ &= \delta^2\sum_{i=1}^{R}\sum_{j,l=1,j\neq l}^{R} s_{ij}s_{il}\mathbf{q}_i\mathbf{q}_i^\top\end{aligned} \tag{45}$$

We now obtain $S_i = \sum_{j,l=1,j\neq l}^{R} s_{ij}s_{il}$. It is easy to obtain that $S_i = 1 - \sum_{j=1}^{R} s_{ij}^2$ because $\mathbf{s}_i$ is a probability simplex and $S_i + \sum_{j=1}^{R} s_{ij}^2 = \left(\sum_{j=1}^{R} s_{ij}\right)^2 = 1$. Thus we have

$$\begin{aligned}S_i &= 1 - \sum_{j=1}^{R} s_{ij}^2 \\ &= 1 - \left(k\cdot p_1^2 + (R-k)\cdot p_0^2\right) \\ &= \frac{k-1}{k} + \frac{2(R-k)}{k}p_0 - \frac{R(R-k)}{k}p_0^2 \\ &\leq \mathcal{O}\left(1 - R^{-\alpha}\right).\end{aligned} \tag{46}$$

Therefore we have

$$\begin{aligned}\mathcal{I}_1 &\leq \delta^2\left(\sum_{i=1}^{R} S_i^2\right)^{1/2}\left(\sum_{i=1}^{R}\mathbf{q}_i\mathbf{q}_i^\top\right)^{1/2} \\ &\leq \mathcal{O}\left(\delta^2 D_0\left(R^{\frac{1}{2}} - R^{\frac{1}{2}-\alpha}\right)\right)\end{aligned} \tag{47}$$

Based on $\mathbb{E}[\mathbf{q}_i\mathbf{q}_j^\top] = 0, \forall i,j \in [R], i \neq j$, we have $\mathcal{I}_2 = 0$. Thus we have

$$\mathbb{E}[[\Delta]_{ab}] = \mathcal{O}\left(\delta^2 D_0\left(R^{\frac{1}{2}} - R^{\frac{1}{2}-\alpha}\right)\right). \tag{48}$$

Hence we have

$$\begin{aligned}\mathbb{E}\|\Delta\|_F \leq \mathbb{E}\|\Delta\|_1 = \mathbb{E}\left[\sum_{a,b=1}^{D}|[\Delta]_{ab}|\right] &\leq \sum_{a,b=1}^{D}\mathbb{E}[|[\Delta]_{ab}|] \\ &= \mathcal{O}\left(\delta^2 D_0 D^2 R^{\frac{1}{2}}\left(1 - R^{-\alpha}\right)\right).\end{aligned} \tag{49}$$

□

## F Deferred Proofs in Section C.2

*Proof of Theorem C.2.* By the $L$-Lipschitz smoothness of $\mathcal{L}$, we have

$$\mathcal{L}(\mathbf{W}^{t+1}) \leq \mathcal{L}(\mathbf{W}^t) + \left\langle\nabla\mathcal{L}(\mathbf{W}^t), \mathbf{W}^{t+1} - \mathbf{W}^t\right\rangle_F + \frac{L}{2}\left\|\mathbf{W}^{t+1} - \mathbf{W}^t\right\|_F^2. \tag{50}$$

We show that

$$\begin{aligned}\left\langle\nabla\mathcal{L}(\mathbf{W}^t), \mathbf{W}^{t+1} - \mathbf{W}^t\right\rangle_F &= -\eta\,\mathrm{Tr}\left(\nabla\mathcal{L}(\mathbf{W}^t)^\top\nabla\mathcal{L}(\mathbf{W}^t)(\mathbf{H}^t)^{-\gamma}\right) \\ &= -\eta\,\mathrm{Tr}\left(\nabla\mathcal{L}(\mathbf{W}^t)(\mathbf{H}^t)^{-\gamma}\nabla\mathcal{L}(\mathbf{W}^t)^\top\right)\end{aligned} \tag{51}$$

and

$$\left\|\mathbf{W}^{t+1} - \mathbf{W}^t\right\|_F^2 = \eta^2\,\mathrm{Tr}\left(\nabla\mathcal{L}(\mathbf{W}^t)(\mathbf{H}^t)^{-\gamma}(\mathbf{H}^t)^{-\gamma}\nabla\mathcal{L}(\mathbf{W}^t)^\top\right). \tag{52}$$

Because $\mathbf{H}^t$ is positive symmetric, we define $\mathbf{H}^t = (\mathbf{V}^t)^\top \Sigma^t \mathbf{V}^t$ and $\tilde{\mathbf{G}}^t = \nabla\mathcal{L}(\mathbf{W}^t)(\mathbf{V}^t)^\top$, we have

$$\begin{aligned}\mathcal{L}(\mathbf{W}^{t+1}) &\leq \mathcal{L}(\mathbf{W}^t) - \eta \operatorname{Tr}\left(\tilde{\mathbf{G}}^t(\Sigma^t)^{-\gamma}(\tilde{\mathbf{G}}^t)^\top\right) + \frac{L\eta^2}{2}\operatorname{Tr}\left(\tilde{\mathbf{G}}^t(\Sigma^t)^{-2\gamma}(\tilde{\mathbf{G}}^t)^\top\right) \\ &= \mathcal{L}(\mathbf{W}^t) - \eta \operatorname{Tr}\left((\tilde{\mathbf{G}}^t)^\top\tilde{\mathbf{G}}^t\left((\Sigma^t)^{-\gamma} - \frac{L\eta}{2}(\Sigma^t)^{-2\gamma}\right)\right).\end{aligned} \tag{53}$$

Let $\mathbf{M} = (\Sigma^t)^{-\gamma} - \frac{L\eta}{2}(\Sigma^t)^{-2\gamma}$. To make sure $\mathbf{M}$ is semi-positive, we need

$$(\lambda_i^t)^{-\gamma}\left(1 - \frac{L\eta}{2}(\lambda_i^t)^{-\gamma}\right) \geq 0, \tag{54}$$

for all $i \in [d]$, where $\lambda_i^t$ is the $i$-th eigenvalue of $\mathbf{H}^t$, which introduces

$$\eta \leq \frac{2(\lambda_i^t)^\gamma}{L}. \tag{55}$$

Denoting $\eta = \frac{\min_{t=0}^T \min_{i=1}^d (\lambda_i^t)^\gamma}{L}$, we have $\mathbf{M} \succeq 0$ and $\eta(\lambda_i^t)^{-\gamma} \leq 1/L$ for all $i \in [d]$ and $t \in [T]$, which implies $1 - \frac{L\eta}{2}(\lambda_i^t)^{-\gamma} \geq \frac{1}{2}$. Thus we have

$$\begin{aligned}\mathcal{L}(\mathbf{W}^{t+1}) &\leq \mathcal{L}(\mathbf{W}^t) - \eta \min_{i=1}^d \left((\lambda_i^t)^{-\gamma}\left(1 - \frac{L\eta}{2}(\lambda_i^t)^{-\gamma}\right)\right)\left\|\tilde{\mathbf{G}}^t\right\|_F^2 \\ &\leq \mathcal{L}(\mathbf{W}^t) - \frac{\eta}{2}\min_{i=1}^d (\lambda_i^t)^{-\gamma}\left\|\nabla\mathcal{L}(\mathbf{W}^t)\right\|_F^2 \\ &= \mathcal{L}(\mathbf{W}^t) - \frac{\eta}{2\max_{i=1}^d (\lambda_i^t)^\gamma}\left\|\nabla\mathcal{L}(\mathbf{W}^t)\right\|_F^2.\end{aligned} \tag{56}$$

Considering the $\mu$-PL condition we have $\|\nabla\mathcal{L}(\mathbf{W}^t)\|_F^2 \geq 2\mu\left(\mathcal{L}(\mathbf{W}^t) - \mathcal{L}^*\right)$. Thus, we have

$$\begin{aligned}\mathcal{L}(\mathbf{W}^{t+1}) - \mathcal{L}^* &\leq \left(1 - \frac{\mu\eta}{\max_{i=1}^d (\lambda_i^t)^\gamma}\right)\left(\mathcal{L}(\mathbf{W}^t) - \mathcal{L}^*\right) \\ &\leq \left(1 - \frac{\mu}{L\tilde{\kappa}_\gamma}\right)\left(\mathcal{L}(\mathbf{W}^t) - \mathcal{L}^*\right),\end{aligned} \tag{57}$$

where $\tilde{\kappa}_\gamma = \frac{\max_{t=0}^T \max_{i=1}^d (\lambda_i^t)^\gamma}{\min_{t=0}^T \min_{i=1}^d (\lambda_i^t)^\gamma}$. It implies that

$$\mathcal{L}(\mathbf{W}^T) - \mathcal{L}^* \leq \exp\left(-\frac{T}{\tilde{\kappa}_\gamma\kappa}\right)\left(\mathcal{L}(\mathbf{W}^0) - \mathcal{L}^*\right). \tag{58}$$

Besides, based on $\mathbf{H}^t = (\mathbf{U}^t)^\top\mathbf{U}^t + \epsilon\mathbf{I} \in \mathbb{R}^{d\times d}$ and $d \leq D$, we have $\lambda_i^t = (\sigma_i^t)^2 + \epsilon$ for all $i \in [d]$ and $t \in [T]$. We finish the proof. □

# G Additional Results and Experiments

## G.1 Experimental Setup and Hyperparameter Selection

**Models and training protocol.** We pre-train LLaMA-style models at three scales, approximately 130M, 350M, and 1.3B parameters, on the C4/en dataset (Raffel et al., 2020). The LLaMA-130M and LLaMA-350M models are trained for 20k and 60k optimizer steps, corresponding to approximately 5.2B and 15.6B training tokens, respectively, while the LLaMA-1.3B model is trained for 100k steps on approximately 40B tokens. For the 130M and 350M experiments, we use a total batch size of 256 with 2k and 6k warmup steps, respectively. For the 1.3B experiments, we use a total batch size of 512 with 10k warmup steps. All experiments use gradient clipping with norm 1.0, a cosine learning-rate schedule decaying to zero, and weight decay 0.1.

**Optimizer configurations.** For AdamW (Loshchilov & Hutter, 2017), we use $(\beta_1, \beta_2) = (0.9, 0.95)$. For Muon, we use momentum $0.95$, while parameters outside the matrix parameter groups are updated by AdamW with the same $(\beta_1, \beta_2)$ values. The remaining Muon configurations follow the commonly used setup in Wen et al. (2025). For STRATA, we use Muon as the base optimizer $\mathcal{A}$ for parameters other than the Query and Key projections. Therefore, the corresponding Muon learning rate and AdamW fallback learning rate are shared with the Muon baseline at the same model scale. Unless otherwise specified, we set the preconditioning order to $\gamma = 1/2$ and use the adaptive damping described in Section 3.3. For the 1.3B experiment, we use N-STRATA, which follows the same optimizer configuration as STRATA while additionally applying the column-wise normalization described in Appendix G.2.

**Learning-rate selection.** We tune the learning rate separately for each model scale while keeping the remaining training configuration fixed. For AdamW, we search over $\{6 \times 10^{-4}, 1 \times 10^{-3}, 1 \times 10^{-4}\}$. For Muon, we report the learning rate as an ordered pair $(\eta_{\mathrm{mat}}, \eta_{\mathrm{fb}})$, where $\eta_{\mathrm{mat}}$ is applied to matrix parameters handled by Muon and $\eta_{\mathrm{fb}}$ is the AdamW learning rate for the remaining parameters. The candidate learning rates used at each model scale are summarized in Table 2. STRATA uses Muon as the base optimizer $\mathcal{A}$ and adopts the corresponding selected Muon configuration for the remaining parameters.

Table 2: Candidate learning rates used in the hyperparameter search. Muon learning rates are reported as $(\eta_{\mathrm{mat}}, \eta_{\mathrm{fb}})$, where $\eta_{\mathrm{mat}}$ is applied to matrix parameters and $\eta_{\mathrm{fb}}$ is the AdamW fallback learning rate for the remaining parameters. Bold indicates the selected configuration.

| Model | Optimizer | Candidate learning rates |
|---|---|---|
| LLaMA-130M | AdamW | $\{1 \times 10^{-4}, \mathbf{6 \times 10^{-4}}, 1 \times 10^{-3}\}$ |
| | Muon | $\{(8 \times 10^{-3}, 1.6 \times 10^{-3}), (\mathbf{1.6 \times 10^{-2}}, \mathbf{3.2 \times 10^{-3}}), (3.2 \times 10^{-2}, 6.4 \times 10^{-3})\}$ |
| LLaMA-350M | AdamW | $\{1 \times 10^{-4}, \mathbf{6 \times 10^{-4}}, 1 \times 10^{-3}\}$ |
| | Muon | $\{(1.6 \times 10^{-3}, 4.8 \times 10^{-4}), (\mathbf{8 \times 10^{-3}}, \mathbf{2.4 \times 10^{-3}}), (1.6 \times 10^{-2}, 4.8 \times 10^{-3})\}$ |
| LLaMA-1.3B | AdamW | $\{1 \times 10^{-4}, \mathbf{6 \times 10^{-4}}, 1 \times 10^{-3}\}$ |
| | Muon | $\{(1 \times 10^{-3}, 1.6 \times 10^{-4}), (\mathbf{5 \times 10^{-3}}, \mathbf{8 \times 10^{-4}}), (1 \times 10^{-2}, 1.6 \times 10^{-3})\}$ |

## G.2 Ablations on the STRATA Design

### G.2.1 Right-sided versus Bilateral Preconditioning

Section 3.1 identifies a bilateral curvature structure for the Query and Key projections. The practical STRATA update retains only the Attention-specific right factor. To examine whether explicitly incorporating the input-dependent left factor further improves optimization, we construct a bilateral variant of STRATA.

Specifically, the bilateral variant additionally applies the left factor $\mathbb{E}[\mathbf{X}^\top \mathbf{X}]$ from Lemma 3.1, while keeping the right preconditioner constructed by STRATA unchanged. For each model scale, all remaining training configurations are identical to those used in the corresponding main experiment, so the only difference is whether the left preconditioner is included.

As shown in Figure 3, the right-sided formulation consistently achieves lower training and validation losses than the bilateral variant on both LLaMA-130M and LLaMA-350M. Thus, under the tested settings, additionally incorporating the input-dependent left factor does not improve convergence. This result supports the practical design adopted by STRATA, which focuses on the architecture-specific right factor while avoiding the additional activation-dependent covariance required by bilateral preconditioning.

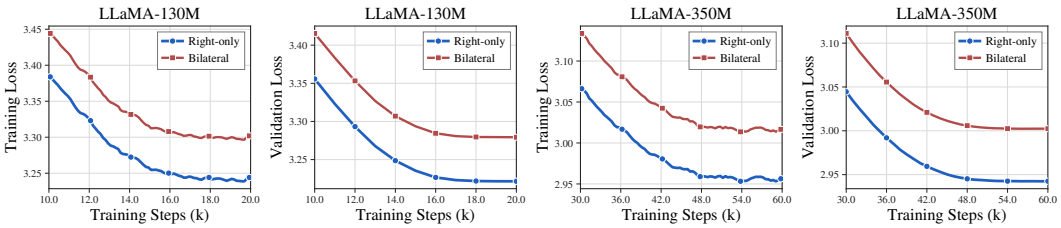
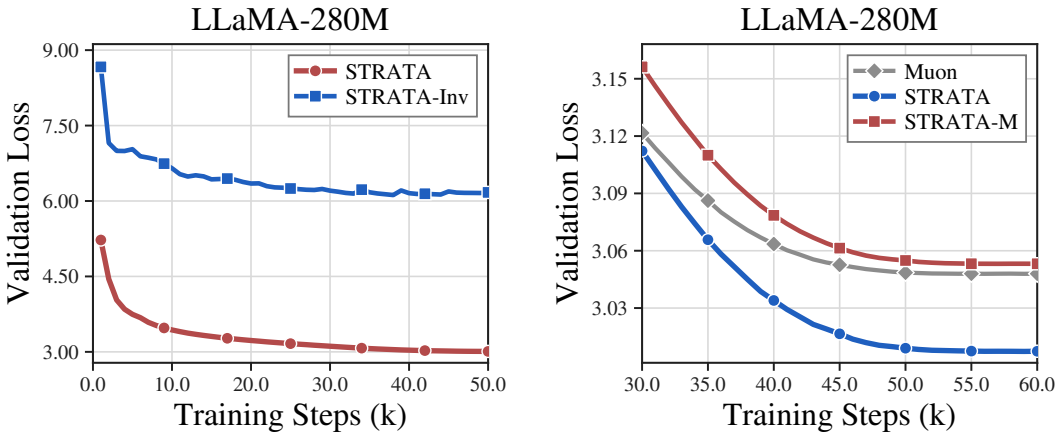



Figure 3: Comparison between right-sided STRATA and its bilateral variant on LLaMA-130M and LLaMA-350M pre-training. The bilateral variant additionally applies the input-dependent left factor from Lemma 3.1, while keeping the STRATA right preconditioner unchanged. All other training configurations follow the corresponding main experiments.

### G.2.2 Choice of the Preconditioning Order

We examine the effect of the preconditioning order $\gamma$ by comparing $\gamma = 1$ and $\gamma = 1/2$. The former applies the full inverse of the constructed preconditioner, whereas the latter uses inverse-square-root preconditioning, which provides a milder correction of the gradient geometry.

To make the effect of the Attention-specific update more visible, we conduct this ablation on a customized 280M-parameter LLaMA model with a relatively larger fraction of Query and Key parameters. As shown in Figure 4 (a), the update with $\gamma = 1$ exhibits substantially worse optimization behavior, while $\gamma = 1/2$ decreases the validation loss smoothly throughout training. We therefore use $\gamma = 1/2$ as the default preconditioning order in the main experiments.




(a) Choice of the preconditioning order

(b) Effect of momentum

Figure 4: Ablation studies on the customized LLaMA-280M model. (a) Comparison between the default STRATA with $\gamma = 1/2$ and the full-inverse variant with $\gamma = 1$, denoted as STRATA-Inv in the legend. (b) Comparison among the Muon baseline, default STRATA, and STRATA with momentum $0.95$ applied before structural preconditioning, denoted as STRATA-M in the legend. Muon is used as the base optimizer $\mathcal{A}$ in both experiments.

### G.2.3 Effect of Momentum

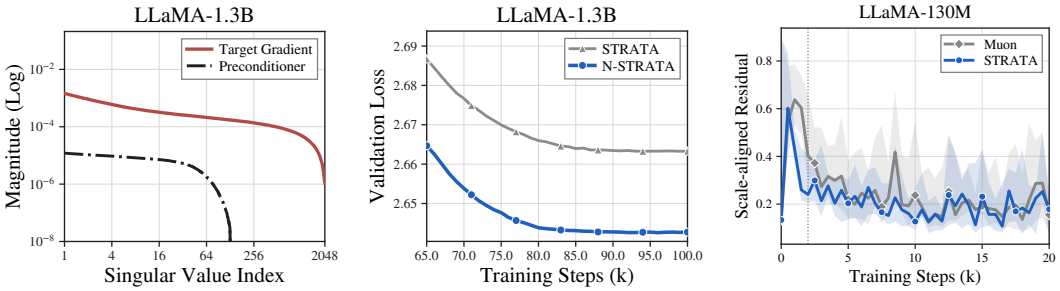
We next examine whether an additional momentum mechanism benefits the STRATA update. In this ablation, the base optimizer is Muon, and we compare the default STRATA update with a variant that applies momentum $0.95$ to the Query and Key gradients before structural preconditioning. All remaining training configurations are kept unchanged.

Figure 4 (b) shows that adding momentum does not improve the optimization trajectory and results in a slightly higher validation loss near the end of training. We therefore apply STRATA directly to the current Query and Key gradients without introducing an additional momentum term in the structural preconditioning branch.

### G.2.4 Effect of Column-wise Normalization

We examine the effect of column-wise normalization at the 1.3B model scale by comparing the standard STRATA update with N-STRATA, which additionally normalizes each column of the preconditioned update. All other training configurations are kept unchanged.

As shown in Figure 5 (b), N-STRATA achieves a lower validation loss and exhibits more stable optimization than the unnormalized STRATA variant on LLaMA-1.3B. This result indicates that column-wise normalization is beneficial for stabilizing the structure-aware update at this larger model scale. We therefore use N-STRATA for the 1.3B experiments reported in the main text.




(a) Spectrum (b) Validation loss v.s. steps (c) Activation-free residual

Figure 5: (a) The visualized spectrum of the gradient and preconditioner introduced by STRATA on $\mathbf{W}_K$ for pre-training LLaMA-1.3B. (b) Comparison between STRATA and N-STRATA on LLaMA-1.3B pre-training. (c) Scale-aligned residual between the activation-dependent target and the activation-free surrogate during LLaMA-130M pre-training.

### G.2.5 Activation-Free Approximation in Practice

A key practical step in STRATA is to eliminate the activation-dependent quantity required by the structural curvature and construct the preconditioner using the Value parameter and its gradient. As derived in Section 3.2, the activation-dependent target $\mathcal{K}_{\text{target}}$ is replaced by the surrogate $\mathcal{K}_{\text{approx}}$, which can be computed from $\mathbf{W}_V$ and $\nabla\mathcal{L}(\mathbf{W}_V)$ without maintaining additional activation-dependent statistics.

To examine this approximation during actual pre-training, we measure the discrepancy between $\mathcal{K}_{\text{target}}$ and $\mathcal{K}_{\text{approx}}$ along the optimization trajectory. Since the two matrices may differ in their overall scale, we report the scale-aligned residual $\min_{c>0} \|\mathcal{K}_{\text{target}} - c\mathcal{K}_{\text{approx}}\|_{\text{F}} / \|\mathcal{K}_{\text{target}}\|_{\text{F}}$. As shown in Figure 5(c), after an initial transient, the residual remains consistently controlled for both Muon and STRATA throughout training. This indicates that the surrogate preserves the dominant matrix geometry of the activation-dependent target up to an overall scaling difference, providing empirical support for the activation-free construction used in STRATA.

### G.2.6 Choice of the Damping Parameter

STRATA adds a damping term to the structural preconditioner to ensure positive definiteness and numerical stability. In our implementation, we set the damping adaptively as $\epsilon = \tilde{\alpha}\,\mathrm{tr}(\mathbf{H})$, where $\mathbf{H}$ denotes the corresponding structural preconditioner before damping. A larger $\epsilon$ increases numerical stability but also weakens the anisotropic effect of the preconditioner by shifting its eigenvalues toward a common scale.

Unless otherwise specified, we set $\tilde{\alpha} = 0.01$ in all experiments. This choice provides a small positive shift relative to the overall scale of the preconditioner while preserving its spectral structure.

## G.3 Robustness across Random Seeds

We further examine whether the optimization advantage of STRATA is sensitive to the random seed. Using the same LLaMA-130M pre-training setup as in the main experiment, we run both Muon and STRATA with three random seeds: 42, 43, and 44. Table 3 reports the final validation losses.

STRATA achieves a lower final validation loss than Muon for all three random seeds. Averaged across the three runs, STRATA obtains $3.2401 \pm 0.0142$, compared with $3.2556 \pm 0.0025$ for Muon. These results indicate that the optimization improvement of STRATA is consistently observed across the tested random seeds, although the magnitude of the gain varies across runs.

Table 3: Final validation loss across three random seeds on LLaMA-130M. Mean and standard deviation are computed over the three runs.

| Optimizer | Seed 42 | Seed 43 | Seed 44 | Mean $\pm$ Std |
|---|---|---|---|---|
| Muon | 3.2575 | 3.2572 | 3.2521 | $3.2556 \pm 0.0025$ |
| STRATA | **3.2215** | **3.2558** | **3.2431** | $\mathbf{3.2401 \pm 0.0142}$ |

### G.4 Downstream Tasks Evaluation

We further evaluate whether the faster pre-training progress of STRATA is accompanied by competitive downstream performance. We conduct zero-shot evaluation on eight standard benchmarks covering commonsense reasoning, scientific knowledge, and linguistic understanding. Specifically, we consider PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), and COPA (Wang et al., 2019) for commonsense reasoning; ARC-e (Chollet, 2019), ARC-c (Moskvichev et al., 2023), and SciQ (Welbl et al., 2017) for scientific knowledge; and ReCoRD (Zhang et al., 2018) and WiC (Pilehvar & Camacho-Collados, 2019) for linguistic understanding. We report results for LLaMA-350M and LLaMA-1.3B, where the models exhibit non-trivial zero-shot performance across these tasks.

Table 4: Zero-shot performance of **LLaMA-350M**. Best results are shown in **bold**.

| Optimizer | PIQA | HellaSwag | ARC-e | ARC-c | COPA | SciQ | ReCoRD | WiC | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| AdamW | 64.15 | 32.94 | **39.10** | 23.12 | 58.00 | 61.00 | 47.13 | 50.16 | 46.95 |
| Muon | **64.80** | 33.81 | 38.89 | **24.91** | 63.00 | **61.90** | 49.15 | 45.77 | 47.78 |
| **STRATA** | **64.80** | **33.97** | 38.38 | 23.55 | **64.00** | 61.40 | **52.95** | **51.25** | **48.79** |

Table 5: Zero-shot performance of **LLaMA-1.3B**. Best results are shown in **bold**.

| Optimizer | PIQA | HellaSwag | ARC-e | ARC-c | COPA | SciQ | ReCoRD | WiC | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| AdamW | 68.28 | 40.75 | 43.18 | **25.68** | 64.00 | 65.00 | 51.38 | 50.16 | 51.05 |
| Muon | 68.55 | 42.97 | 43.48 | 24.91 | **68.00** | **66.40** | 55.03 | 45.77 | 51.89 |
| **N-STRATA** | **68.77** | **43.07** | **43.86** | 25.34 | 67.00 | 66.10 | **56.44** | **51.25** | **52.73** |

As shown in Tables 4 and 5, (N-)STRATA achieves the highest average zero-shot score at both model scales. On LLaMA-350M, STRATA improves the average score from $47.78$ with Muon to $48.79$, while on LLaMA-1.3B, N-STRATA improves it from $51.89$ to $52.73$. The gains are not uniform across individual tasks: Muon or AdamW remains better on several benchmarks, whereas (N-)STRATA performs particularly well on ReCoRD and WiC.

These results indicate that the faster pre-training convergence observed in the main experiments does not come at the expense of downstream zero-shot performance. Instead, across the tested model scales, (N-)STRATA maintains competitive performance on individual benchmarks while improving the average score relative to the AdamW and Muon baselines.

## H Convergence of Coupled Newton-Schulz Iteration

We introduce the convergence analysis of the coupled Newton-Schulz iteration for obtaining the $-1/2$ power of the given positive symmetric matrix $\mathbf{A}$ in Algorithm 2. We start from a standard Newton-Schulz iteration. In order to obtain $\mathbf{A}^{-1/2}$, we construct

$$F(\mathbf{X}) = \mathbf{X}^{-2} - \mathbf{A}, \tag{59}$$

and try to find a zero point of $F(\mathbf{X})$. Taking the first-order Taylor expansion of $F$ at $\mathbf{X}^t$, we have

$$F(\mathbf{X}) \approx F(\mathbf{X}^t) + F'(\mathbf{X}^t)(\Delta\mathbf{X}). \tag{60}$$

Letting $F(\mathbf{X}) = 0$ and $\Delta\mathbf{X} = \mathbf{X}^{t+1} - \mathbf{X}^t$, we have

$$(\mathbf{X}^t)^{-2} - \mathbf{A} + F'(\mathbf{X}^t)\left(\mathbf{X}^{t+1} - \mathbf{X}^t\right) = 0. \tag{61}$$

Letting $\mathbf{X}^0$ and $\mathbf{A}$ commutative, (e.g., $\mathbf{X}^0 = \mathbf{I}$), based on mathematical induction, all the $\{\mathbf{X}^t\}$ and $\mathbf{A}$ is commutative. Therefore, we deal with $F'(\mathbf{X})$ as scalar function. We have

$$(\mathbf{X}^t)^{-2} - \mathbf{A} - 2(\mathbf{X}^t)^{-3}\left(\mathbf{X}^{t+1} - \mathbf{X}^t\right) = 0, \tag{62}$$

which implies

$$\mathbf{X}^{t+1} = \frac{1}{2}\mathbf{X}^t\left(3\mathbf{I} - \mathbf{A}(\mathbf{X}^t)^2\right). \tag{63}$$

Denoting $\mathbf{R}^t = \mathbf{I} - (\mathbf{X}^t)^2\mathbf{A}$ as the error, if $\mathbf{X}^t = \mathbf{A}^{-1/2}$, we have $\mathbf{R}^t = 0$. Considering $\mathbf{R}^{t+1}$, letting $\mathbf{M}^t = (\mathbf{X}^t)^2\mathbf{A}$, we have

$$\begin{aligned}\mathbf{M}^{t+1} &= \left(\frac{\mathbf{X}^t}{2}(3\mathbf{I} - \mathbf{M}^t)\right)^2\mathbf{A} \\ &= \mathbf{M}^t\left(\frac{(3\mathbf{I} - \mathbf{M}^t)^2}{4}\right) \\ &= \frac{9}{4}\mathbf{M}^t - \frac{3}{2}(\mathbf{M}^t)^2 + \frac{1}{4}(\mathbf{M}^t)^3.\end{aligned} \tag{64}$$

Applying $\mathbf{R}^t = \mathbf{I} - \mathbf{M}^t$ and $\mathbf{R}^{t+1} = \mathbf{I} - \mathbf{M}^{t+1}$, we have

$$\mathbf{I} - \mathbf{R}^{t+1} = \mathbf{I} - \frac{3}{4}(\mathbf{R}^t)^2 - \frac{1}{4}(\mathbf{R}^t)^3, \tag{65}$$

which implies

$$\mathbf{R}^{t+1} = \mathcal{O}\left((\mathbf{R}^t)^2\right). \tag{66}$$

That means the iteration has a quadratic convergence speed.

The coupled Newton-Schulz iteration is equivalent to the above standard one. Because $\mathbf{X}^0 = \mathbf{A}$ and $\mathbf{Y}^0 = \mathbf{I}$ is the polynomial of $\mathbf{A}$, we have $\mathbf{X}^t\mathbf{Y}^t = \mathbf{Y}^t\mathbf{X}^t$ for all $t$. That means

$$\mathbf{Y}^{t+1} = \mathbf{Y}^t\mathbf{Z}^t. \tag{67}$$

Considering $\mathbf{X}^0 = \mathbf{A}\mathbf{Y}^0$, supposing $\mathbf{X}^t = \mathbf{A}\mathbf{Y}^t$, we have

$$\mathbf{X}^{t+1} = \mathbf{X}^t\mathbf{Z}^t = \mathbf{A}\mathbf{Y}^t\mathbf{Z}^t = \mathbf{A}\mathbf{Y}^{t+1}. \tag{68}$$

Thus we have for all $t$, $\mathbf{X}^t = \mathbf{A}\mathbf{Y}^t$, which implies

$$\mathbf{Y}^{t+1} = \mathbf{Y}^t\mathbf{Z}^t = \frac{1}{2}\mathbf{Y}^t(3\mathbf{I} - \mathbf{A}(\mathbf{Y}^t)^2), \tag{69}$$

which is equivalent to (63). Thus $\mathbf{Y}^t$ converges to $\mathbf{A}^{-1/2}$ with a quadratic convergence speed.

The sufficient condition of convergence is $\rho(\mathbf{I} - \mathbf{Y}^0\mathbf{X}^0) < 1$, where $\rho$ is spectral radius. That introduces $\|\mathbf{I} - \mathbf{Y}^{t+1}\mathbf{X}^{t+1}\| \leq \|\mathbf{I} - \mathbf{Y}^t\mathbf{X}^t\|^2$. To ensure this condition, we conduct a normalization before running iterations in Algorithm 2. $\mathbf{Y}^{T_1} \approx (\mathbf{A}/\|\mathbf{A}\|_F)^{-1/2} = \mathbf{A}^{-1/2}\|\mathbf{A}\|_F^{1/2}$, so we rescale it after the iterations.

# I Deferred Proofs in Appendix A

*Proof of Proposition A.2.* We begin by deriving the matrix-form expansion aggregated over the input sequence. For each token $i$, the pre-activation vector is $\mathbf{W}^\top\mathbf{a}_i \in \mathbb{R}^d$. Under the perturbation $\mathbf{W} \mapsto \mathbf{W} - \mathbf{Q}$, the pre-activation changes by $-\mathbf{Q}^\top\mathbf{a}_i$. By the integral remainder form of Taylor's theorem, the expansion for the $i$-th token is given by:

$$\mathcal{L}_i(\mathbf{W}^\top\mathbf{a}_i - \mathbf{Q}^\top\mathbf{a}_i) = \mathcal{L}_i(\mathbf{W}^\top\mathbf{a}_i) - \nabla\mathcal{L}_i(\mathbf{W}^\top\mathbf{a}_i)^\top(\mathbf{Q}^\top\mathbf{a}_i) + h_i(\mathbf{Q}^\top\mathbf{a}_i, \mathbf{W}^\top\mathbf{a}_i),$$

where the exact quadratic remainder $h_i$ is defined by the path-dependent integral:

$$h_i(\mathbf{Q}^\top\mathbf{a}_i, \mathbf{W}^\top\mathbf{a}_i) := (\mathbf{Q}^\top\mathbf{a}_i)^\top\left[\int_0^1 (1-t)\nabla^2\mathcal{L}_i(\mathbf{W}^\top\mathbf{a}_i - t\mathbf{Q}^\top\mathbf{a}_i)\mathrm{d}t\right](\mathbf{Q}^\top\mathbf{a}_i).$$

Based on our assumption, we approximate this path-dependent integral with the shared feature-dimension curvature matrix $\frac{1}{2}\mathbf{H}$. Substituting this directly into the remainder term yields:

$$h_i(\mathbf{Q}^\top \mathbf{a}_i, \mathbf{W}^\top \mathbf{a}_i) \approx \frac{1}{2}(\mathbf{Q}^\top \mathbf{a}_i)^\top \mathbf{H}(\mathbf{Q}^\top \mathbf{a}_i).$$

Consequently, the token-level Taylor expansion simplifies to an approximation parameterized by $\mathbf{H}$:

$$\mathcal{L}_i(\mathbf{W}^\top \mathbf{a}_i - \mathbf{Q}^\top \mathbf{a}_i) \approx \mathcal{L}_i(\mathbf{W}^\top \mathbf{a}_i) - \nabla \mathcal{L}_i(\mathbf{W}^\top \mathbf{a}_i)^\top (\mathbf{Q}^\top \mathbf{a}_i) + \frac{1}{2}(\mathbf{Q}^\top \mathbf{a}_i)^\top \mathbf{H}(\mathbf{Q}^\top \mathbf{a}_i).$$

Next, we aggregate this token-level expansion to construct the global loss $\mathcal{L}(\mathbf{W} - \mathbf{Q}) = \frac{1}{R}\sum_{i=1}^{R} \mathcal{L}_i(\mathbf{W}^\top \mathbf{a}_i - \mathbf{Q}^\top \mathbf{a}_i)$. Summing the terms over all $R$ tokens and taking the average, we obtain:

$$\mathcal{L}(\mathbf{W} - \mathbf{Q}) \approx \frac{1}{R}\sum_{i=1}^{R} \mathcal{L}_i(\mathbf{W}^\top \mathbf{a}_i) - \frac{1}{R}\sum_{i=1}^{R} \nabla \mathcal{L}_i(\mathbf{W}^\top \mathbf{a}_i)^\top (\mathbf{Q}^\top \mathbf{a}_i) + \frac{1}{2R}\sum_{i=1}^{R} (\mathbf{Q}^\top \mathbf{a}_i)^\top \mathbf{H}(\mathbf{Q}^\top \mathbf{a}_i).$$

Recognizing that the first term reconstructs the original global loss $\mathcal{L}(\mathbf{W})$ and the first-order summation corresponds to the trace inner product with the global gradient $\mathrm{tr}(\mathbf{Q}^\top \mathbf{G})$, we apply the trace property $\mathbf{x}^\top \mathbf{H}\mathbf{x} = \mathrm{tr}(\mathbf{H}\mathbf{x}\mathbf{x}^\top)$ to the second-order term:

$$\mathcal{L}(\mathbf{W} - \mathbf{Q}) \approx \mathcal{L}(\mathbf{W}) - \mathrm{tr}(\mathbf{Q}^\top \mathbf{G}) + \frac{1}{2R}\mathrm{tr}\left(\mathbf{H}\mathbf{Q}^\top \left(\sum_{i=1}^{R} \mathbf{a}_i \mathbf{a}_i^\top\right)\mathbf{Q}\right).$$

Since the sum of the outer products of the token inputs recovers the sequence covariance matrix, $\sum_{i=1}^{R} \mathbf{a}_i \mathbf{a}_i^\top = \mathbf{A}^\top \mathbf{A}$, the global matrix-form Taylor expansion is established as:

$$\mathcal{L}(\mathbf{W} - \mathbf{Q}) \approx \mathcal{L}(\mathbf{W}) - \mathrm{tr}(\mathbf{Q}^\top \mathbf{G}) + \frac{1}{2R}\mathrm{tr}(\mathbf{H}\mathbf{Q}^\top \mathbf{A}^\top \mathbf{A}\mathbf{Q}).$$

Besides, by flattening the weight matrix into a vector $\mathrm{vec}(\mathbf{W}) \in \mathbb{R}^{Dd}$, the standard second-order Taylor expansion of the global loss $\mathcal{L}$ around $\mathrm{vec}(\mathbf{W})$ under a perturbation $-\mathbf{Q}$ gives:

$$\mathcal{L}(\mathbf{W} - \mathbf{Q}) \approx \mathcal{L}(\mathbf{W}) - \mathrm{tr}(\mathbf{Q}^\top \mathbf{G}) + \frac{1}{2}\mathrm{vec}(\mathbf{Q})^\top \mathcal{H}_{\mathbf{W}} \mathrm{vec}(\mathbf{Q}).$$

We now try to analyze the relationship of the second-order term in the two Taylor expansions above. We first analyze the exact second-order change for token-coupled losses. Let the pre-activation vector at token $i$ be $\mathbf{s}_i = \mathbf{W}^\top \mathbf{a}_i \in \mathbb{R}^d$ for $i = 1, \ldots, R$. We stack these token outputs into a single vector $\mathbf{s} := [\mathbf{s}_1^\top, \ldots, \mathbf{s}_R^\top]^\top \in \mathbb{R}^{Rd}$, and let the averaged global scalar loss be $\mathcal{L}(\mathbf{s}) := L(\mathbf{s})/R$, where $L(\mathbf{s}) = \sum_{i=1}^{R} \mathcal{L}_i(\mathbf{s}_i)$. The exact token-coupled curvature of $L$ is given by the block Hessian matrix:

$$\mathscr{H} := \nabla_{\mathbf{s}}^2 L(\mathbf{s}) \in \mathbb{R}^{(Rd)\times(Rd)}.$$

We can write $\mathscr{H}$ in $R \times R$ blocks of size $d \times d$:

$$\mathscr{H} = \begin{bmatrix} \mathbf{H}_{11} & \cdots & \mathbf{H}_{1R} \\ \vdots & \ddots & \vdots \\ \mathbf{H}_{R1} & \cdots & \mathbf{H}_{RR} \end{bmatrix}, \quad \mathbf{H}_{ij} \in \mathbb{R}^{d\times d}.$$

For an update direction $\mathbf{Q} \in \mathbb{R}^{D\times d}$, the pre-activation perturbation at token $i$ is $-\mathbf{Q}^\top \mathbf{a}_i$. Based on the exact output-space curvature, the full second-order change of the global loss is:

$$\delta^2 \mathcal{L} = \frac{1}{2R}\sum_{i=1}^{R}\sum_{j=1}^{R} (\mathbf{Q}^\top \mathbf{a}_i)^\top \mathbf{H}_{ij} (\mathbf{Q}^\top \mathbf{a}_j).$$

To expose the Kronecker structure, we utilize the trace formulation $(\mathbf{Q}^\top \mathbf{a}_i)^\top \mathbf{H}_{ij}(\mathbf{Q}^\top \mathbf{a}_j) = \mathrm{tr}(\mathbf{H}_{ij}\mathbf{Q}^\top \mathbf{a}_j \mathbf{a}_i^\top \mathbf{Q})$. Applying the standard Kronecker product vectorization identity $\mathrm{tr}(\mathbf{C}\mathbf{X}^\top \mathbf{B}\mathbf{X}) = \mathrm{vec}(\mathbf{X})^\top (\mathbf{B}^\top \otimes \mathbf{C})\mathrm{vec}(\mathbf{X})$ with $\mathbf{X} = \mathbf{Q}$, $\mathbf{B} = \mathbf{a}_j \mathbf{a}_i^\top$, and $\mathbf{C} = \mathbf{H}_{ij}$, each term can be rewritten in the parameter space:

$$(\mathbf{Q}^\top \mathbf{a}_i)^\top \mathbf{H}_{ij}(\mathbf{Q}^\top \mathbf{a}_j) = \mathrm{vec}(\mathbf{Q})^\top \Big((\mathbf{a}_i \mathbf{a}_j^\top) \otimes \mathbf{H}_{ij}\Big)\mathrm{vec}(\mathbf{Q}).$$

Extracting the quadratic form with respect to $\mathrm{vec}(\mathbf{Q})$, the exact parameter-space Hessian in the vector-based Taylor expansion is:

$$\mathcal{H}_{\mathbf{W}} = \frac{1}{R}\sum_{i=1}^{R}\sum_{j=1}^{R}(\mathbf{a}_i\mathbf{a}_j^\top) \otimes \mathbf{H}_{ij} \in \mathbb{R}^{(Dd)\times(Dd)}. \tag{70}$$

Equ. (70) represents the exact, intractable parameter-space Hessian. To make this practically useful, we define the diagonal-block average curvature $\mathbf{H} := \frac{1}{R}\sum_{i=1}^{R}\mathbf{H}_{ii} \in \mathbb{R}^{d\times d}$. It is a standard simplifying approximation that assumes negligible cross-token curvature, foundational to methods like K-FAC which retain only the diagonal-token curvature contributions. It yields:

$$\mathcal{H}_{\mathbf{W}} \approx \frac{1}{R}\sum_{i=1}^{R}(\mathbf{a}_i\mathbf{a}_i^\top) \otimes \mathbf{H} = \frac{1}{R}(\mathbf{A}^\top\mathbf{A}) \otimes \mathbf{H}.$$

This completes the proof.

□