DOES U-W RATIO ALIGNMENT BENEFIT MUON IN LLM PRETRAINING?

Anonymous authors
Paper under double-blind review

ABSTRACT

Most top-performing entries in the `modded-nanogpt` optimization benchmark use a "u/w floor". The u/w floor lower-bounds the Frobenius-norm ratio of each block's pre-learning-rate update to its weight (*u-w ratio*). Like decoupled weight decay, the u/w floor prevents updates from becoming negligible relative to the current weights during training. Unlike decoupled weight decay, however, the u/w floor also aligns u-w ratios across blocks. Specifically, it aligns the u-w ratios of most weight blocks in a model architecture to a common value on most training steps. We therefore ask whether this u-w ratio alignment is an underlying mechanism of the u/w floor's empirical advantage. To study this question, we remove the u/w floor and directly enforce u-w ratio alignment on Muon updates. This gives *MuonL*, which assigns a common ratio to all Muon-updated blocks, and *typed MuonL (t-MuonL)*, which aligns only blocks with the same Transformer role. To assess these policies independently of particular learning-rate values, we derive a measurable local diagnostic from a theoretical model of one-step loss decrease. When all Muon-updated blocks share the same learning rate, the analysis yields a covariance-based sufficient condition for when MuonL improves upon Muon. When the learning rates can vary across matrix types, it yields a generalized-eigenvalue criterion for when t-MuonL outperforms Muon and vice versa. We then evaluate the diagnostic at checkpoints along Muon training trajectories, using each checkpoint as a starting point to compare Muon's one-step decrease with the counterfactual decreases under MuonL and t-MuonL. It favors t-MuonL over Muon in both cases and MuonL over Muon in the former. Controlled runs using the same training configurations corroborate the diagnostic, with lower final validation losses for MuonL or t-MuonL than for Muon. These results support u-w ratio alignment as a contributing factor to the u/w floor's empirical advantage and a useful intervention for Muon-based LLM pretraining.

1 INTRODUCTION

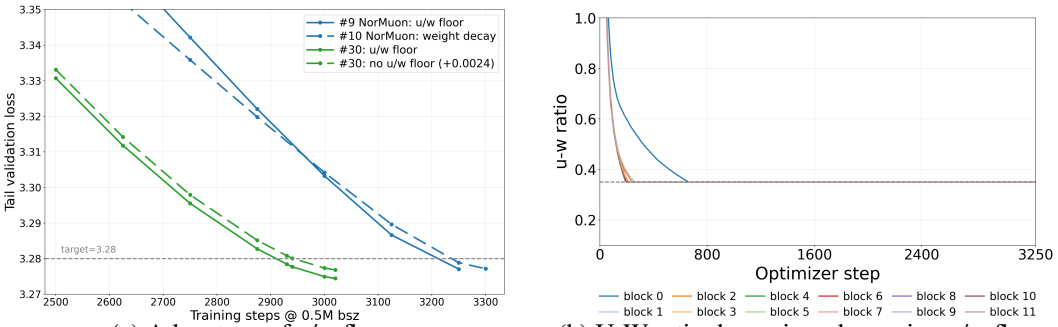
Muon has recently emerged as a strong optimizer for large language model (LLM) training (Jordan et al., 2024b; Liu et al., 2025; Shah et al., 2025). Its empirical strength has drawn substantial attention and led to a rapidly growing family of Muon-style variants (Du & Su, 2026; Kimi Team, 2025; Zhang et al., 2026; Si et al., 2025). This development is reflected in the recent `modded-nanogpt` optimization benchmark (Track 3), which compares optimizers by the minimum training steps required to reach a target validation loss (Jordan, 2026). The benchmark reveals several techniques that repeatedly improve training efficiency. However, the underlying mechanisms of their effectiveness remain unclear.

One such intervention is the *u/w floor*, which has become a recurring component of the strongest benchmark records. As of September 1, 2026, nine of the ten best entries in the benchmark retain the u/w floor, while the other uses a generalized row-level u/w floor (Jordan, 2026).[1] Moreover, Figure 1(a) summarizes two existing benchmark comparisons in which the runs with the u/w floor reach the target validation loss in about 0.8–1.1% fewer steps than their counterparts without it.

[1]The nine entries are #45, #44, #43, #42, #41, #40, #38, #34, and #30. Here, entry numbers identify submission order rather than performance ranks.

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.

Index the Muon-updated linear transformations in the Transformer by i. Let W_i denote the corresponding weight matrix and U_i the Muon update before applying the learning rate, obtained by uniformizing the singular values of the momentum-adjusted gradient. We measure the relative update scale by the *u-w ratio*, defined as $\|U_i\|_F / \|W_i\|_F$. The u/w floor lower-bounds the u-w ratio at c by rescaling U_i whenever it falls below c. Like decoupled weight decay, this prevents excessively small u-w ratios and hence negligible updates relative to the current weights (Li et al., 2020). However, because the same threshold c is applied for all blocks, the u/w floor additionally induces the alignment of u-w ratios across blocks. Figure 1(b) illustrates this alignment in a NorMuon run following the configuration of entry #9 in Figure 1(a): the u-w ratios of the self-attention output projection matrices W_O decrease to $c = 0.35$ early in training and remain concentrated at that level thereafter. Decoupled weight decay (Loshchilov & Hutter, 2019), in contrast, does not explicitly enforce such alignment at every step. This motivates asking whether u-w ratio alignment across blocks contributes to the u/w floor's empirical advantage.



(a) Advantage of u/w floor.

(b) U-W ratio dynamics when using u/w floor.

Figure 1: **Motivation for u-w ratio alignment.** Entry numbers denote submission order rather than performance rank. **(a)** Advantage of the u/w floor from two comparisons in the `modded-nanogpt` optimization benchmark (Track 3). The blue curves compare two benchmark entries for NorMuon (Li et al., 2026), one of the stronger Muon variants in the benchmark: entry #9 with u/w floor threshold 0.35 (averaged over 8 seeds) versus entry #10 with no u/w floor and a decoupled weight-decay coefficient of 0.025 (averaged over 20 seeds). The solid green curve shows Aurora (Dewulf et al., 2026) with the u/w floor (entry #30), averaged over 16 seeds. The dashed green curve represents the same Aurora implementation with only the u/w floor removed and is reconstructed by shifting the solid curve upward by 0.0024, the validation-loss increase measured at step 2940 over four reruns. **(b)** U-W ratios for the self-attention output projection matrices W_O in a NorMuon run following benchmark entry #9, with u/w floor threshold $c = 0.35$. Each colored curve corresponds to one Transformer block; the horizontal dashed line marks c.

To isolate the effect of u-w ratio alignment, we remove the u/w floor and directly enforce u-w ratio alignment on the Muon updates. This gives *MuonL*[2], which assigns a common u-w ratio to all Muon-updated matrices. MuonL is reasonable only when all Muon-updated blocks share the same learning rate, which we call the *shared-learning-rate setting*; otherwise, the relative update sizes become misaligned after scaling by the learning rates. However, prior work shows LLM pretraining can benefit when learning rates vary across module types, such as query/key projections, value/output projections, and feed-forward layers (Wang et al., 2025; Wen et al., 2026b). We refer to this as the *type-specific learning-rate setting*. To accommodate this setting, we propose *typed MuonL (t-MuonL)*, which aligns u-w ratios only within each matrix type. Using the Transformer architecture in the `modded-nanogpt` benchmark, we define six types by architectural role across Transformer blocks: the self-attention query, key, value, and output projection matrices W_Q, W_K, W_V, and W_O, and the MLP up- and down-projection matrices W_{up} and W_{down} (Vaswani et al., 2017).

Even in the shared-learning-rate setting, t-MuonL has an advantage over MuonL in that it deviates less from Muon's shape scaling.[3] This is desirable because our goal is to enforce u-w ratio alignment while otherwise staying close to Muon, whose shape scaling sets the update norms $\|U_i\|_F$ according to matrix dimensions (Bernstein & Newhouse, 2024). U-W ratio alignment across matrix types can

[2]Though MuonL has a similar core principle to OrScale (vision configuration) (Lou & You, 2026), their algorithms differ. We elaborate on this in Appendix B.

[3]See Appendix A.3 and Figure 7 for supporting evidence.

substantially distort the update norms prescribed by shape scaling when matrix dimensions differ across types. Since matrices within each type have identical dimensions in NanoGPT (Karpathy, 2022), restricting alignment within type reduces this distortion.

These motivations lead to two questions: does u-w ratio alignment across blocks benefit Muon in LLM pretraining, and if so, which alignment granularity is beneficial? In the shared-learning-rate setting, we compare both MuonL and t-MuonL with Muon; when type-specific learning rates are allowed, we compare t-MuonL with Muon.

1.1 Contributions

Building on the questions above, we formulate u-w ratio alignment as a block-wise scaling of fixed Muon update directions in Section 2. In Section 3, we establish a local score based on the optimized one-step loss decrease, followed by interpretable covariance-based criteria for when a u-w ratio alignment policy outperforms Muon. Section 4 evaluates these diagnostics at checkpoints along Muon training trajectories, using each checkpoint as a common starting point for the candidate policies. The framework also applies to analogous alignment policies for other optimizers, as illustrated for SGD in Appendix D. Our contributions are threefold.

- **A local diagnostic for u-w ratio alignment.** We develop measurable local scores and covariance-based criteria, both independent of specific learning-rate values, for assessing when u-w ratio alignment is beneficial for Muon.
- **t-MuonL for type-specific learning-rate settings.** We introduce t-MuonL, a within-type u-w ratio alignment policy compatible with type-specific learning rates. It also reduces the distortion of shape scaling relative to MuonL in shared-learning-rate settings.
- **U-W ratio alignment as a mechanism behind the u/w floor.** By relating the alignment dynamics induced by the u/w floor to our diagnostics and experiments on MuonL, we provide evidence that u-w ratio alignment contributes to the u/w floor's empirical advantage.

1.2 Background and Related Work

The `modded-nanogpt` optimization benchmark. NanoGPT provides a lightweight framework for training GPT-style language models (Karpathy, 2022), and `modded-nanogpt` builds on it as a collaborative benchmark for efficient GPT pretraining (Jordan et al., 2024a). Its optimization track holds the architecture, data, and batch size fixed, and evaluates an optimizer by the statistically validated number of training steps required to reach a target validation loss (Jordan, 2026). This controlled protocol isolates optimizer behavior from systems-level throughput and enables direct comparison of optimizer interventions.

Connection between decoupled weight decay and U-W ratio. Prior work shows that weight decay can modulate the effective learning rate and prevent the relative update scale $\|U\|_F/\|W\|_F$ from becoming too small, particularly in scale-invariant networks (van Laarhoven, 2017; Li et al., 2020; Bjorck et al., 2021). Moreover, for SGD and scale-invariant parameters under suitable conditions, decoupled weight decay yields an approximate equilibrium u-w ratio of $\sqrt{2\lambda/\eta}$, where η is the learning rate and λ is the weight-decay coefficient (Chiley et al., 2019; Wan et al., 2021). Under Muon, Wen et al. (2026a) further show that $\mathbb{E}\|U_t\|_F^2/\mathbb{E}\|W_t\|_F^2$ reaches a shape-dependent equilibrium. Thus, same-shaped blocks converge to the same second-moment u-w ratio. Since blocks within each matrix type have the same shape, t-MuonL can be viewed as enforcing this within-type alignment before this equilibrium is reached.

U-W ratio control. Several previous works indirectly control and align the u-w ratio. The Hyperball wrapper fixes the relative update to a constant before projection onto the hyperball $\{W : \|W\|_F = R\}$ (Wen et al., 2026a); consequently, the u-w ratios are approximately aligned across blocks when the updates are mostly tangential. The Rotational Optimizer removes the radial component of each update and controls its magnitude using an exponential moving average of past update scales; its u-w ratios are therefore aligned only through these running RMS scales, rather than explicitly equalized across blocks at each step (Kosson et al., 2024). Our work instead directly analyzes stepwise u-w ratio alignment, rescaling the Muon updates while preserving their directions.

Trust-ratio alignment. Trust-ratio alignment is equivalent to u-w ratio alignment in principle. LARS introduces this mechanism by multiplying each block's update by a factor proportional to $\|W_i\|_F/\|U_i\|_F$, thereby equalizing relative update sizes across blocks (You et al., 2017). LAMB and OrScale (vision configuration) incorporate this trust-ratio scaling into Adam- and Muon-style updates, respectively (You et al., 2020; Lou & You, 2026). However, their alignment mechanisms have primarily been demonstrated in BERT and CNN training. Our work is most closely related to this line of research; Appendix B elaborates on the detailed differences.

2 SETUP

We first set up a theoretical framework for analyzing u-w ratio alignment for a general optimizer. Let $\mathcal{I}$ index the matrix parameter blocks updated by the optimizer of interest. In our `modded-nanogpt` setting, $\mathcal{I}$ indexes the weight matrices within Transformer blocks. For each $i \in \mathcal{I}$, let W_i denote the weight matrix and U_i the matrix produced by the optimizer to update W_i before scaling by the learning rate η_i. Define the original u-w ratio $r_i := \|U_i\|_F/\|W_i\|_F$ and the unit-Frobenius update direction $V_i := U_i/\|U_i\|_F$. In the analysis, we assume $\|W_i\|_F > 0$ and $\|U_i\|_F > 0$. A u-w ratio alignment policy replaces r_i by some aligned ratio x_i while preserving the direction V_i. Thus we can write the parameter change as

$$\Delta_i(x) := -\eta_i x_i \|W_i\|_F V_i, \tag{1}$$

so different u-w ratio alignment policies differ only in the assigned ratios $x = (x_i)_{i\in\mathcal{I}}$. Throughout the diagnostic analysis, we omit the decoupled weight-decay term to isolate the effect of u-w ratio alignment on the Muon updates and keep the resulting criteria more interpretable. In the actual training algorithms and experiments, Muon, MuonL, and t-MuonL all use the same decoupled weight decay (which is compatible with u-w ratio alignment) to prevent overly small relative updates.

We then partition $\mathcal{I}$ into N matrix types $\mathfrak{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_N\}$, with each type containing matrices that serve the same role across Transformer layers. In our setting, the six types are the query, key, value, and output projection matrices W_Q, W_K, W_V, W_O, and the MLP up- and down-projection matrices W_{up} and W_{down}. Let $\mathcal{G}(i)$ denote the type containing block i, and $|\mathcal{G}|$ the number of blocks in type $\mathcal{G}$. The learning-rate setting specifies η_i in Equation 1. Under a shared learning rate, write $\eta_i = \eta$ for every block. Under type-specific learning rates, we write $\eta_i = \eta_{\mathcal{G}(i)}$.

Under this formulation, Muon leaves each original ratio r_i unchanged, t-MuonL assigns a common ratio within each matrix type, and MuonL assigns a common ratio across all blocks. They therefore correspond to no alignment, within-type alignment, and full alignment across all blocks, respectively. In the notation of Equation 1, these policies are given by

$$x_i^{\text{base}} = r_i, \quad x_i^{\text{typed}} = \left(\tfrac{1}{|\mathcal{G}(i)|}\textstyle\sum_{j\in\mathcal{G}(i)} r_j^{-1}\right)^{-1}, \quad x_i^{\text{full}} = \left(\tfrac{1}{|\mathcal{I}|}\textstyle\sum_{j\in\mathcal{I}} r_j^{-1}\right)^{-1}, \tag{2}$$

so that each alignment group is assigned the harmonic mean of its original u-w ratios. Algorithms 1 and 2 in Appendix A.2 detail the implementations of MuonL and t-MuonL.

For the shared-learning-rate setting, we compare all three policies using a score $\mathcal{S}_{\text{SL}}(x)$ ("SL" stands for "shared learning rate"), which measures the local *optimization potential* across all blocks in $\mathcal{I}$, i.e., the optimized one-step loss decrease under the model developed in Section 3. With type-specific learning rates, full alignment does not generally equalize the relative parameter change $\|\Delta_i(x^{\text{full}})\|_F/\|W_i\|_F = \eta_{\mathcal{G}(i)} x_i^{\text{full}}$, which still depends on the learning-rate magnitude. We therefore compare only alignment policies x^{typed} and x^{base} in this setting. The analysis is based on scores $\mathcal{S}_{\mathcal{G}}(x)$ for the optimization potential within each type $\mathcal{G}$, and $\mathcal{S}_{\text{TL}}(x)$ ("TL" stands for "type-specific learning rate") for the optimization potential across all Muon-updated matrix blocks indexed by $\mathcal{I}$.

3 THEORY

We first construct the scores introduced in Section 2 by maximizing the local quadratic model of one-step loss decrease over the learning rate. Based on these scores, we further derive covariance-based criteria for when u-w ratio alignment improves Muon's local optimization potential.

3.1 LOCAL DIAGNOSTIC ANALYSIS

Recall that $\mathcal{I}$ indexes the Muon-updated matrices. Let $W = (W_i)_{i\in\mathcal{I}}$ denote these matrices and $\mathcal{L}(W)$ the training loss as a function of W, with all remaining model parameters held fixed. Assume that $\mathcal{L}$ is twice continuously differentiable in a neighborhood of W. Write

$$w_i := \|W_i\|_F, \qquad G_i := \nabla_{W_i}\mathcal{L}(W), \qquad H_{ij} := \nabla^2_{W_i,W_j}\mathcal{L}(W),$$

for the weight norm, block gradient, and block Hessian operator, respectively. For a generic parameter change $\Delta = (\Delta_i)_{i\in\mathcal{I}}$ (see Equation 1), write $\|\Delta\|_F^2 := \sum_{i\in\mathcal{I}} \|\Delta_i\|_F^2$ for its product-space Frobenius norm. Taylor's theorem then gives

$$\mathcal{L}(W+\Delta) = \mathcal{L}(W) + \sum_{i\in\mathcal{I}} \langle G_i, \Delta_i\rangle_F + \frac{1}{2}\sum_{i,j\in\mathcal{I}} \langle \Delta_i, H_{ij}[\Delta_j]\rangle_F + o(\|\Delta\|_F^2). \tag{3}$$

Define

$$a_i := \langle G_i, V_i\rangle_F, \qquad C_{ij} := \langle V_i, H_{ij}[V_j]\rangle_F.$$

Here a_i measures first-order descent along $-V_i$, C_{ii} is the curvature along that direction, and C_{ij} for $i \neq j$ captures the Hessian interaction between blocks i and j.

Shared-learning-rate score. With a shared learning rate $\eta_i = \eta$ for all matrix indices $i \in \mathcal{I}$, substituting $\Delta(x) = (\Delta_i(x))_{i\in\mathcal{I}}$ from Equation 1 into Equation 3 gives

$$\mathcal{L}(W) - \mathcal{L}(W+\Delta(x)) = \eta A(x) - \frac{\eta^2}{2}B(x) + o(\|\Delta(x)\|_F^2), \tag{4}$$

where $A(x) := \sum_{i\in\mathcal{I}} x_i w_i a_i$ and $B(x) := \sum_{i,j\in\mathcal{I}} x_i x_j w_i w_j C_{ij}$ are the first-order and curvature terms, respectively. When $B(x) > 0$, the quadratic part is maximized over the learning rate $\eta \in \mathbb{R}$[4] at $\eta^*(x) = A(x)/B(x)$, giving the optimal quadratic score

$$\mathcal{S}_{\mathrm{SL}}(x) := \frac{A(x)^2}{2B(x)}, \tag{5}$$

which measures the local optimization potential with policy x. Recall that r_i and C_{ij} stand for the u-w ratio and Hessian interaction. Define the *adjusted directional curvature*

$$\widetilde{L}_i^{\mathcal{I}} := C_{ii} + \sum_{\substack{j\in\mathcal{I}\\ j\neq i}} \frac{2r_j w_j}{w_i(r_i+r_j)} C_{ij}, \tag{6}$$

which combines within-block curvature with weighted cross-block interactions across $\mathcal{I}$. Substituting x^{full} and x^{base} into Equation 5, with first-order descent coefficients $a_i = \langle G_i, V_i\rangle_F$, yields

$$\mathcal{S}_{\mathrm{SL}}(x^{\mathrm{full}}) = \frac{\left(\sum_{i\in\mathcal{I}} w_i a_i\right)^2}{2\sum_{i\in\mathcal{I}} \widetilde{L}_i^{\mathcal{I}} w_i^2}, \qquad \mathcal{S}_{\mathrm{SL}}(x^{\mathrm{base}}) = \frac{\left(\sum_{i\in\mathcal{I}} r_i w_i a_i\right)^2}{2\sum_{i\in\mathcal{I}} \widetilde{L}_i^{\mathcal{I}} r_i^2 w_i^2}. \tag{7}$$

For within-type alignment, write $\bar{r}_{\mathcal{G}} := \left(|\mathcal{G}|^{-1}\sum_{i\in\mathcal{G}} r_i^{-1}\right)^{-1}$ for the harmonic-mean u-w ratio within type $\mathcal{G} \in \mathfrak{G}$, where $\mathfrak{G}$ is the matrix-type partition from Section 2. Substituting into Equation 5 gives

$$\mathcal{S}_{\mathrm{SL}}(x^{\mathrm{typed}}) = \frac{\left(\sum_{\mathcal{G}\in\mathfrak{G}} \bar{r}_{\mathcal{G}} \sum_{i\in\mathcal{G}} w_i a_i\right)^2}{2\sum_{\mathcal{G},\mathcal{H}\in\mathfrak{G}} \bar{r}_{\mathcal{G}}\bar{r}_{\mathcal{H}} \sum_{i\in\mathcal{G}}\sum_{j\in\mathcal{H}} w_i w_j C_{ij}}. \tag{8}$$

Within-type score. To assess u-w ratio alignment within a single matrix type, we consider updating only the blocks in a prespecified matrix type $\mathcal{G}$, using a common learning rate $\eta_{\mathcal{G}}$. Restrict the first-order term $A(x)$, curvature term $B(x)$, and adjusted directional curvature $\widetilde{L}_i^{\mathcal{I}}$ to $\mathcal{G}$, and denote the resulting quantities by $A_{\mathcal{G}}(x)$, $B_{\mathcal{G}}(x)$, and $\widetilde{L}_i^{\mathcal{G}}$. When $B_{\mathcal{G}}(x) > 0$, maximizing over the type learning rate $\eta_{\mathcal{G}} \in \mathbb{R}$ gives the following score for the optimization potential within type $\mathcal{G}$:

$$\mathcal{S}_{\mathcal{G}}(x) := \frac{A_{\mathcal{G}}(x)^2}{2B_{\mathcal{G}}(x)}. \tag{9}$$

[4]Throughout the analysis, the learning rate η or the entries of the learning-rate vector $\boldsymbol{\eta}$ may take signed values to characterize the quadratic-model optimum, not to prescribe the actual learning rates used in training.

The scores $\mathcal{S}_{\mathcal{G}}(x^{\text{typed}})$ and $\mathcal{S}_{\mathcal{G}}(x^{\text{base}})$ have the forms in Equation 7, with $\mathcal{I}$ replaced by $\mathcal{G}$ and full alignment replaced by within-type alignment. Because $\mathcal{S}_{\mathcal{G}}$ excludes cross-type curvature interactions, improving it for every type does not guarantee a higher optimization potential $\mathcal{S}_{\text{TL}}$ (analyzed in Section 3.3) when all matrix types are updated together.

3.2 MUON-SPECIFIC COVARIANCE CRITERION

We next give covariance-based sufficient conditions for MuonL to improve the shared-learning-rate score $\mathcal{S}_{\text{SL}}$ and for t-MuonL to improve the within-type score $\mathcal{S}_{\mathcal{G}}$, each relative to Muon. The following definition is useful for stating these conditions.

Definition 1 (Weighted normalization and covariance) *Let $\mathcal{J}$ be a finite index set and $\pi = (\pi_i)_{i\in\mathcal{J}}$ be probability weights. For scalar collections $f = (f_i)_{i\in\mathcal{J}}$ and $g = (g_i)_{i\in\mathcal{J}}$, define the weighted mean and normalization by*

$$\langle f\rangle_\pi := \sum_{i\in\mathcal{J}} \pi_i f_i, \qquad [f_i]_\pi := \frac{f_i}{\langle f\rangle_\pi}$$

whenever $\langle f\rangle_\pi \neq 0$, and define the corresponding weighted covariance and variance by

$$\operatorname{cov}_\pi(f,g) := \langle fg\rangle_\pi - \langle f\rangle_\pi\langle g\rangle_\pi, \qquad \operatorname{var}_\pi(f) := \operatorname{cov}_\pi(f,f).$$

For uniform weights $\pi_i = |\mathcal{J}|^{-1}$, we use $\mathcal{J}$ instead of π as the subscript for unweighted averaging, normalization, covariance, and variance.

For Muon, let m_i and n_i denote the row and column dimensions of block i, set $k_i := \min\{m_i, n_i\}$, and let M_i be the momentum matrix orthogonalized by Muon, assumed to have full rank. Given its compact SVD $M_i = P_i\Sigma_i Q_i^\top$, define $\operatorname{msgn}(M_i) := P_i Q_i^\top$. Following Keller Jordan's shape scaling (Jordan et al., 2024b), the pre-learning-rate update is

$$U_i = \sqrt{\max\{1, m_i/n_i\}}\,\operatorname{msgn}(M_i).$$

Recall from Section 3.1 that V_i is the unit-Frobenius update direction, a_i is the first-order loss-decrease coefficient along $-V_i$, and r_i is the original u-w ratio. Since $\|\operatorname{msgn}(M_i)\|_F = \sqrt{k_i}$, for Muon these quantities are

$$V_i = \frac{\operatorname{msgn}(M_i)}{\sqrt{k_i}}, \qquad a_i = \frac{\langle G_i, \operatorname{msgn}(M_i)\rangle_F}{\sqrt{k_i}}, \qquad r_i = \frac{\sqrt{m_i}}{w_i}.$$

These Muon identities yield the following sufficient conditions for t-MuonL and MuonL to improve upon Muon.

Proposition 1 (Sufficient criteria for (t-)MuonL) *Let m_i and n_i denote the row and column dimensions of block i, and set $k_i = \min\{m_i, n_i\}$. Assume the within-type curvature terms $B_{\mathcal{G}}(x^{\text{typed}}), B_{\mathcal{G}}(x^{\text{base}}) > 0$, with first-order term $A_{\mathcal{G}}(x^{\text{base}}) \neq 0$. If*

$$2\operatorname{cov}_{\mathcal{G}}\left([w_i]_{\mathcal{G}}, [a_i]_{\mathcal{G}}\right) - \operatorname{cov}_{\mathcal{G}}\left(([w_i]_{\mathcal{G}})^2, [\widetilde{L}_i^{\mathcal{G}}]_{\mathcal{G}}\right) > \operatorname{var}_{\mathcal{G}}([w_i]_{\mathcal{G}}), \tag{10}$$

then $\mathcal{S}_{\mathcal{G}}(x^{\text{typed}}) > \mathcal{S}_{\mathcal{G}}(x^{\text{base}})$, i.e., t-MuonL outperforms Muon in optimization potential within type $\mathcal{G}$. Let $\pi_i = m_i/\sum_{j\in\mathcal{I}} m_j$ for $i \in \mathcal{I}$. Assume the curvature terms $B(x^{\text{full}}), B(x^{\text{base}}) > 0$, with first-order term $A(x^{\text{base}}) \neq 0$. If

$$2\operatorname{cov}_\pi\left(\left[\frac{w_i}{\sqrt{m_i}}\right]_\pi, \left[\frac{a_i}{\sqrt{m_i}}\right]_\pi\right) - \operatorname{cov}_\pi\left(\left[\frac{w_i}{\sqrt{m_i}}\right]_\pi^2, [\widetilde{L}_i^{\mathcal{I}}]_\pi\right) > \operatorname{var}_\pi\left(\left[\frac{w_i}{\sqrt{m_i}}\right]_\pi\right), \tag{11}$$

then $\mathcal{S}_{\text{SL}}(x^{\text{full}}) > \mathcal{S}_{\text{SL}}(x^{\text{base}})$, i.e., MuonL outperforms Muon in optimization potential under a shared learning rate.

We refer to the three terms in Equation 10, from left to right, as the *first-order descent gain*, *curvature gain*, and *rescaling cost*, respectively; Equation 11 uses the same terminology for the corresponding terms. Assuming all first-order entries a_i to be positive, the within-type criterion favors t-MuonL over Muon in terms of optimization potential within $\mathcal{G}$ when matrices with larger weight norms in $\mathcal{G}$ also exhibit greater first-order descent and lower adjusted directional curvature, provided these gains outweigh the rescaling cost. The comparison between MuonL and Muon across all Muon-updated matrices follows a similar tradeoff, but involves dimension-adjusted weight norms and first-order descent terms.

3.3 TYPE-SPECIFIC LEARNING-RATE SETTING

We now compare t-MuonL with Muon when all matrix types are updated using type-specific learning rates $\boldsymbol{\eta} = (\eta_{\mathcal{G}})_{\mathcal{G}\in\mathfrak{G}}$. Recall from Section 3.1 the weight norms w_i, first-order terms a_i, and directional Hessian interactions C_{ij}. For matrix types $\mathcal{G}, \mathcal{H} \in \mathfrak{G}$, define the first-order vector $\mathbf{A}(x) \in \mathbb{R}^N$ and curvature matrix $\mathbf{B}(x) \in \mathbb{R}^{N\times N}$, where N is the number of matrix types in $\mathfrak{G}$, with entries indexed by $\mathcal{G}$ and $(\mathcal{G}, \mathcal{H})$, respectively, given by

$$A_{\mathcal{G}}(x) := \sum_{i\in\mathcal{G}} x_i w_i a_i, \qquad B_{\mathcal{G},\mathcal{H}}(x) := \sum_{i\in\mathcal{G}}\sum_{j\in\mathcal{H}} x_i x_j w_i w_j C_{ij}, \qquad \mathcal{G},\mathcal{H} \in \mathfrak{G}.$$

Substituting the type-specific parameter changes into the local quadratic model gives the quadratic predicted decrease $\boldsymbol{\eta}^\top \mathbf{A}(x) - \frac{1}{2}\boldsymbol{\eta}^\top \mathbf{B}(x)\boldsymbol{\eta}$ (see Appendix C.1 for the derivation). If $\mathbf{B}(x)$ is positive definite, maximizing this quantity over $\boldsymbol{\eta} \in \mathbb{R}^N$ gives

$$\mathcal{S}_{\mathrm{TL}}(x) := \frac{1}{2}\mathbf{A}(x)^\top \mathbf{B}(x)^{-1}\mathbf{A}(x). \tag{12}$$

Muon-specific comparison. Within each type $\mathcal{G}$, matrices share row and column dimensions, so $k_i = \min\{m_i, n_i\}$ is constant for $i \in \mathcal{G}$; denote it by $k_{\mathcal{G}}$. We further define notation

$$\begin{aligned} g_{\mathcal{G}} &:= \langle a_i\rangle_{\mathcal{G}}, & K_{\mathcal{G},\mathcal{H}} &:= \langle C_{ij}\rangle_{\mathcal{G}\times\mathcal{H}}, \\ c_{\mathcal{G}} &:= \mathrm{cov}_{\mathcal{G}}\left([w_i]_{\mathcal{G}}, [a_i]_{\mathcal{G}}\right), & \mathbf{D} &:= \mathrm{diag}(1 + c_{\mathcal{G}_1}, \ldots, 1 + c_{\mathcal{G}_N}). \end{aligned} \tag{13}$$

Let $\mathbf{g} = (g_{\mathcal{G}})_{\mathcal{G}\in\mathfrak{G}}$ and $\mathbf{K} = (K_{\mathcal{G},\mathcal{H}})_{\mathcal{G},\mathcal{H}\in\mathfrak{G}}$. Here the average over $\mathcal{G}\times\mathcal{H}$ includes all pairs $i \in \mathcal{G}$ and $j \in \mathcal{H}$, including $i = j$ when $\mathcal{G} = \mathcal{H}$. Define the symmetric matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{N\times N}$ with entries

$$\boldsymbol{\Gamma}_{\mathcal{G},\mathcal{H}} := (c_{\mathcal{G}} + c_{\mathcal{H}} + c_{\mathcal{G}}c_{\mathcal{H}})K_{\mathcal{G},\mathcal{H}} - \mathrm{cov}_{\mathcal{G}\times\mathcal{H}}([w_i]_{\mathcal{G}}[w_j]_{\mathcal{H}}, C_{ij}). \tag{14}$$

Using these abstractions, we can derive the following criterion for whether t-MuonL results in higher optimization potential than Muon under type-specific learning rates.

Proposition 2 (Criterion for t-MuonL with type-specific learning rates) *Assume that* $\mathbf{K}$ *and* $\mathbf{DKD} - \boldsymbol{\Gamma}$ *are positive definite, with* $g_{\mathcal{G}} \neq 0$ *and* $c_{\mathcal{G}} \neq -1$ *for every type* $\mathcal{G}$*. Let* $\{(\rho_k, \mathbf{v}_k)\}_{k=1}^N$ *be the generalized eigenpairs of the matrix pair* $(\mathbf{D}^{-1}\boldsymbol{\Gamma}\mathbf{D}^{-1}, \mathbf{K})$*, with* $\rho_k \in \mathbb{R}$ *and* $\mathbf{v}_k \in \mathbb{R}^N$ *satisfying*

$$\mathbf{D}^{-1}\boldsymbol{\Gamma}\mathbf{D}^{-1}\mathbf{v}_k = \rho_k \mathbf{K}\mathbf{v}_k, \qquad \mathbf{v}_k^\top \mathbf{K}\mathbf{v}_\ell = \mathbf{1}\{k = \ell\}.$$

Then

$$\mathcal{S}_{\mathrm{TL}}(x^{\mathrm{typed}}) > \mathcal{S}_{\mathrm{TL}}(x^{\mathrm{base}}) \quad \Longleftrightarrow \quad \sum_{k=1}^{N} \frac{\rho_k}{1-\rho_k}(\mathbf{v}_k^\top \mathbf{g})^2 > 0. \tag{15}$$

In other words, t-MuonL has greater optimization potential than Muon when the sum in Equation 15 is positive, and vice versa. We next interpret each term in this sum. For the k-th generalized eigenmode, choose the learning rates for Muon and t-MuonL so that both have the same first-order decrease $\mathbf{v}_k^\top \mathbf{g}$ (see Equation E.12 in Appendix E.5). Under these choices, ρ_k is t-MuonL's fractional reduction in quadratic curvature relative to Muon. Thus, the k-th term in Equation 15 is the squared matched first-order decrease $(\mathbf{v}_k^\top \mathbf{g})^2$, weighted by $\rho_k/(1-\rho_k)$. Since $\rho_k < 1$, the weight $\rho_k/(1-\rho_k)$ is strictly increasing in ρ_k. Therefore, larger ρ_k implies a larger contribution of the k-th eigenmode to the sum and favors t-MuonL more strongly, while negative ρ_k favors Muon.

Using Equation 14 in the generalized eigenvalue relation (see Appendix C.1) gives

$$\rho_k = \sum_{\mathcal{G},\mathcal{H}\in\mathfrak{G}} v_{k,\mathcal{G}} v_{k,\mathcal{H}} \Big[K_{\mathcal{G},\mathcal{H}} - \frac{K_{\mathcal{G},\mathcal{H}} + q_{\mathcal{G},\mathcal{H}}}{(1+c_{\mathcal{G}})(1+c_{\mathcal{H}})}\Big], \tag{16}$$

where $v_{k,\mathcal{G}}, K_{\mathcal{G},\mathcal{H}}$ are corresponding entries of $\mathbf{v}_k$ and $\mathbf{K}$ and

$$c_{\mathcal{G}} = \mathrm{cov}_{\mathcal{G}}([w_i]_{\mathcal{G}},\, [a_i]_{\mathcal{G}}) \quad , \quad q_{\mathcal{G},\mathcal{H}} := \mathrm{cov}_{\mathcal{G}\times\mathcal{H}}([w_i]_{\mathcal{G}}[w_j]_{\mathcal{H}},\, C_{ij})\,.$$

Thus, the covariance between block weight norms and first-order descent terms a_i, as well as the covariance between pairwise products of block weight norms and directional Hessian interactions C_{ij}, enter the criterion in Equation 15. These two effects are analogous to the first-order descent gain and curvature gain, respectively, in Proposition 1.

4 EXPERIMENTS

4.1 PROTOCOL AND SETUP

We train a 12-layer NanoGPT model on FineWeb10B using the `modded-nanogpt` benchmark configuration (Jordan, 2026). For both learning-rate settings, we evaluate the diagnostics from Section 3 every 250 steps along a 3300-step Muon training trajectory, giving 13 evaluations. At each checkpoint, we compare Muon's optimized one-step decrease with the counterfactual decreases under MuonL or t-MuonL using the same model parameters, block-wise update directions, and diagnostic batch. As in Section 2, we compare all three methods under a shared learning rate and compare t-MuonL with Muon under type-specific learning rates.

In the shared-learning-rate setting, both the diagnostic trajectory and validation runs use the learning-rate configuration from benchmark entry #12, a Muon baseline entry in Track 3. Unless otherwise noted, the diagnostic trajectories and validation runs in this section use seed 42. Appendix A.1 provides the remaining training and diagnostic details, including the learning-rate configuration for the type-specific setting.

4.2 RESULTS

Shared learning rates. The shared-learning-rate score $\mathcal{S}_{\mathrm{SL}}$ in Equation 5 favors t-MuonL over Muon at all 13 evaluation points and MuonL over Muon at 11 of 13 (Figure 2(a)). Figure 2(b) shows the sufficient criterion margin, defined as the left-hand side minus the right-hand side of Equation 11, and its decomposition. The criterion diagnostic certifies that MuonL offers greater local optimization potential than Muon at most states (11 of 13) along the training trajectory. The decomposition shows that, from step 750 onward, the first-order descent gain mainly drives MuonL's advantage, while the typically negative curvature gain partially offsets it; the rescaling cost remains stable.

Under the configuration used for diagnostics, t-MuonL and MuonL achieve lower late-stage validation losses than Muon (Figure 2(c)), consistent with the diagnostic comparisons. Appendix A.3 reports results for additional seeds.

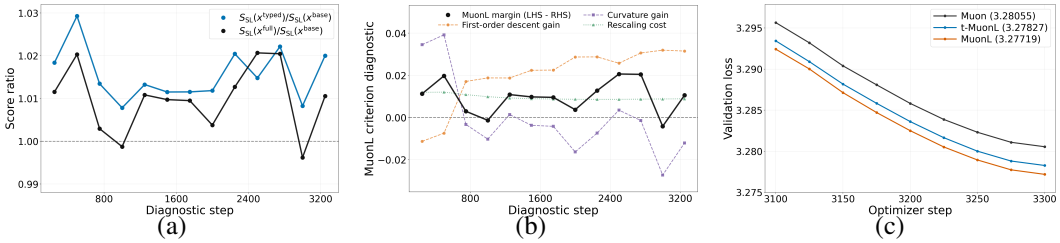


Figure 2: **Shared-learning-rate diagnostics and validation losses.** **(a)** Score ratios $\mathcal{S}_{\mathrm{SL}}(x^{\mathrm{typed}})/\mathcal{S}_{\mathrm{SL}}(x^{\mathrm{base}})$ (t-MuonL compared to Muon, blue) and $\mathcal{S}_{\mathrm{SL}}(x^{\mathrm{full}})/\mathcal{S}_{\mathrm{SL}}(x^{\mathrm{base}})$ (MuonL compared to Muon, black); values above one favor the aligned policy over Muon. **(b)** MuonL sufficient-criterion margin in Equation 11 (black), decomposed into first-order descent gain (orange), curvature gain (purple), and rescaling cost (green). Positive margins favor MuonL. **(c)** Late-stage validation loss for Muon, t-MuonL, and MuonL under the shared-learning-rate configuration.

Type-specific learning rates. We next compare t-MuonL with Muon under type-specific learning rates using the criterion in Proposition 2. The criterion supports an overall advantage for t-MuonL, with substantial positive values early in training outweighing the smaller negative values later (Figure 3(a)). This early advantage is driven mainly by the eigenmode with the second-largest ρ_k at each evaluation, owing to the large squared first-order decrease $(\mathbf{v}_k^\top \mathbf{g})^2$ matched between Muon and t-MuonL.

Figure 3(b) sums the within-type criterion margins from Proposition 1 across the six matrix types and decomposes the sum into first-order descent gain, curvature gain, and rescaling cost. The summed margin remains positive throughout, suggesting an overall advantage for t-MuonL when only within-

type optimization potential is assessed.[5] Unlike the MuonL comparison under a shared learning rate, this positive margin is driven by the curvature gain, while the first-order descent gain is negative.

Under the same configuration used for diagnostics, t-MuonL achieves lower late-stage validation loss than Muon (Figure 3(c)), consistent with the overall advantage indicated by the diagnostic. Validation results across additional seeds are provided in Appendix A.3.

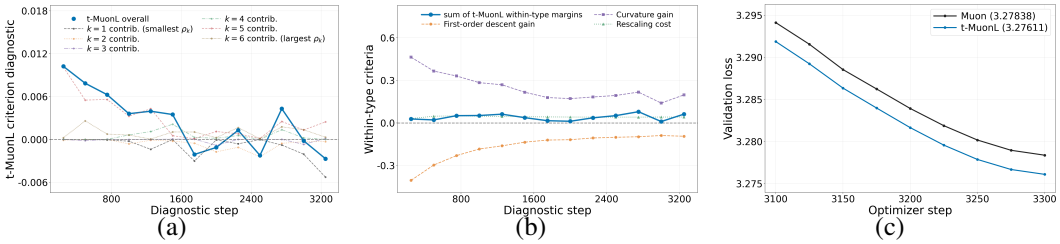


Figure 3: **Type-specific learning-rate diagnostics and validation losses for t-MuonL versus Muon.** **(a)** Criterion sum in Equation 15 (blue) and its six eigenmode contributions, ordered by increasing ρ_k at each evaluation. Positive values favor t-MuonL; negative values favor Muon. **(b)** Sum of the six within-type criterion margins in Equation 10 (blue), with each margin defined as left-hand side minus right-hand side, decomposed into first-order descent gain (orange), curvature gain (purple), and rescaling cost (green). **(c)** Late-stage validation losses for t-MuonL and Muon under the type-specific learning-rate configuration used for diagnostics.

Relation to the u/w floor. Figure 4 compares Muon and MuonL with decoupled weight decay to Muon with the u/w floor under otherwise identical settings across five seeds. MuonL with decoupled weight decay and Muon with the u/w floor achieve similar final validation losses, both lower than Muon with decoupled weight decay. Together with the preceding diagnostics, this suggests that u-w ratio alignment contributes to the u/w floor's empirical advantage.

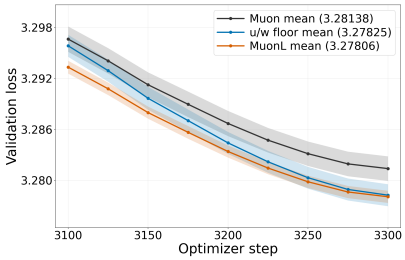


Figure 4: **Validation-loss comparison with the u/w floor.** Muon and MuonL with decoupled weight decay are compared with Muon using the u/w floor (threshold $c = 0.35$)[6], under otherwise identical training settings and five random seeds. Curves show the mean validation loss; shaded bands denote one standard deviation across seeds.

5 Conclusion

In this paper, we develop measurable local scores and covariance-based criteria for assessing u-w ratio alignment in Muon-based LLM pretraining independently of particular learning-rate values. Along Muon training trajectories in the `modded-nanogpt` benchmark, the diagnostics favor t-MuonL over Muon in both learning-rate settings and MuonL over Muon under a shared learning rate. Training runs under the same configurations corroborate the diagnostic results, with MuonL and t-MuonL achieving lower validation losses than Muon in the settings where they are respectively compared. Together with the u-w ratio alignment induced by the u/w floor, these results suggest that the alignment contributes to the u/w floor's performance gains. The framework also applies to other optimizers because the general analysis does not assume a particular update rule.

[5] This comparison serves only as a reference: a positive summed within-type margin does not imply t-MuonL has greater optimization potential than Muon under type-specific learning rates.

[6] This threshold follows benchmark entry #9, a NorMuon run with a u/w floor.

Reproducibility Statement

We provide the information needed to reproduce the theoretical and empirical results of this work. Appendix A provides the experimental setup, implementation details, diagnostic protocol, and multi-seed validation results for the main study. Appendix B provides the comparison with OrScale, including implementation details and supporting experiments. Appendices C and D provide the analysis under type-specific learning rates and the criteria for general optimizers including SGD, while complete proofs are given in Appendix E. Anonymized source code for reproducing the training runs, diagnostics, and figures is included in the supplementary material.

AI Use Statement

In this work, we used generative AI tools to formulate mathematical claims, assist in the writing of proofs, implement methods, assist with translation, and summarize or analyze existing literature. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, or interpret results, and generating synthetic datasets, cleaning or reformatting datasets, and supporting qualitative or thematic data analysis are not applicable to this work. We have reviewed all AI-assisted work. For mathematical claims, generative AI tools were used to carry out and cross-check specific calculations, whose correctness and resulting expressions were subsequently verified by the authors. For proofs, the authors reviewed and checked AI-assisted initial derivations and revised them to improve readability and logical coherence. AI-assisted training and diagnostic code was reviewed and tested by the authors for correctness and consistency with the intended methods. For literature summarization and analysis, the authors manually verified AI-assisted claims against the original sources. During manuscript preparation, generative AI tools were used to assist a native Chinese-speaking author with precise English expression in selected passages. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Anuj Apte. Scale weight decay and train better. *arXiv preprint arXiv:2607.23777*, 2026. URL https://arxiv.org/abs/2607.23777.

Jeremy Bernstein and Laker Newhouse. Modular duality in deep learning. *arXiv preprint arXiv:2410.21265*, 2024. URL https://arxiv.org/abs/2410.21265.

Johan Bjorck, Kilian Q. Weinberger, and Carla Gomes. Understanding decoupled and early weight decay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6777–6785, 2021. doi: 10.1609/aaai.v35i8.16837.

Vitaliy Chiley, Ilya Sharapov, Atli Kosson, Urs Koster, Ryan Reece, Sofia Samaniego de la Fuente, Vishal Subbiah, and Michael James. Online normalization for training neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Alec Dewulf, Dhruv Pai, Li Yang, Ashley Zhang, and Ben Keigwin. Aurora: A leverage-aware spectral optimizer. *arXiv preprint arXiv:2606.27715*, 2026. URL https://arxiv.org/abs/2606.27715.

Zhehang Du and Weijie Su. The Newton-Muon optimizer. *arXiv preprint arXiv:2604.01472*, 2026. URL https://arxiv.org/abs/2604.01472.

Keller Jordan. Modded-NanoGPT optimization benchmark, 2026. URL https://github.com/KellerJordan/modded-nanogpt/tree/5bc4417e842cafac758c5dc276abd9beb7f32612/records/track_3_optimization. Track 3 snapshot accessed September 1, 2026.

Keller Jordan, Jeremy Bernstein, Brendan Rappazzo, @fernbear.bsky.social, Boza Vlado, You Jiacheng, Franz Cesista, Braden Koszarsky, and @Grad62304977. Modded-NanoGPT: Speedrunning the NanoGPT baseline, 2024a. URL https://github.com/KellerJordan/modded-nanogpt.

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

Andrej Karpathy. nanoGPT, 2022. URL https://github.com/karpathy/nanoGPT. GitHub repository.

Kimi Team. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025. URL https://arxiv.org/abs/2507.20534.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Atli Kosson, Bettina Messmer, and Martin Jaggi. Rotational equilibrium: How weight decay balances learning across neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 25333–25369. PMLR, 2024.

Xiang Li, Shuo Chen, and Jian Yang. Understanding the disharmony between weight normalization family and weight decay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4715–4722, 2020. doi: 10.1609/aaai.v34i04.5904.

Zichong Li, Liming Liu, Chen Liang, Weizhu Chen, and Tuo Zhao. NorMuon: Making Muon more efficient and scalable. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is scalable for LLM training. *arXiv preprint arXiv:2502.16982*, 2025. URL https://arxiv.org/abs/2502.16982.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Yuxuan Lou and Yang You. OrScale: Orthogonalized optimization with layer-wise trust-ratio scaling. *arXiv preprint arXiv:2605.07815*, 2026. URL https://arxiv.org/abs/2605.07815.

Ishaan Shah, Anthony M. Polloreno, Karl Stratos, Philip Monk, Adarsh Chaluvaraju, Andrew Hojel, Andrew Ma, Anil Thomas, Ashish Tanwer, Darsh J. Shah, Khoi Nguyen, Kurt Smith, Michael Callahan, Michael Pust, Mohit Parmar, Peter Rushton, Platon Mazarakis, Ritvik Kapila, Saurabh Srivastava, Somanshu Singla, Tim Romanski, Yash Vanjani, and Ashish Vaswani. Practical efficiency of Muon for pretraining. *arXiv preprint arXiv:2505.02222*, 2025. URL https://arxiv.org/abs/2505.02222.

Chongjie Si, Debing Zhang, and Wei Shen. AdaMuon: Adaptive Muon optimizer. *arXiv preprint arXiv:2507.11005*, 2025. URL https://arxiv.org/abs/2507.11005.

Twan van Laarhoven. l_2 regularization versus batch and weight normalization. *arXiv preprint arXiv:1706.05350*, 2017. URL https://arxiv.org/abs/1706.05350.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Ruosi Wan, Zhanxing Zhu, Xiangyu Zhang, and Jian Sun. Spherical motion dynamics: Learning dynamics of normalized neural network using SGD and weight decay. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Jinbo Wang, Mingze Wang, Zhanpeng Zhou, Junchi Yan, Weinan E, and Lei Wu. The sharpness disparity principle in transformers for accelerating language model pre-training. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 64859–64879. PMLR, 2025.

Kaiyue Wen, Xingyu Dang, Kaifeng Lyu, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them II: Hyperball optimization. *arXiv preprint arXiv:2606.16899*, 2026a. URL https://arxiv.org/abs/2606.16899.

Ziqing Wen, Zhouyang Liu, Jiahuan Wang, Ping Luo, Li Shen, Dongsheng Li, and Tao Sun. Revealing modular gradient noise imbalance in LLMs: Calibrating Adam via signal-to-noise ratio. *arXiv preprint arXiv:2605.05794*, 2026b. URL https://arxiv.org/abs/2605.05794.

Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017. URL https://arxiv.org/abs/1708.03888.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training BERT in 76 minutes. In *International Conference on Learning Representations*, 2020.

Ruijie Zhang, Yequan Zhao, Ziyue Liu, Zhengyang Wang, Yupeng Su, Liyan Tan, and Zheng Zhang. MUON+: Towards more effective Muon via one additional normalization step for LLM pretraining. *arXiv preprint arXiv:2602.21545*, 2026. URL https://arxiv.org/abs/2602.21545.

A Experimental Details

A.1 Setting Details

Model and training configuration. Following the `modded-nanogpt` optimization benchmark (Jordan, 2026) (Track 3), we use the 162-M NanoGPT model trained on FineWeb10B. Specifically, it has a 12-layer Transformer architecture, with each Transformer layer containing six Muon-updated matrices: the self-attention query, key, value, and output projection matrices W_Q, W_K, W_V, and W_O, and the MLP up- and down-projection matrices W_{up} and W_{down}. All model and training configurations follow entry #12, a Muon baseline in the benchmark, with the following key configurations and modifications highlighted below:

- **Key configuration 1.** The embedding, output, and non-matrix parameters are updated by AdamW (Loshchilov & Hutter, 2019).
- **Key configuration 2.** We apply 12 Newton-Schulz iterations using the quintic Muon iteration, following the configuration in entry #12.
- **Key configuration 3.** In the shared-learning-rate setting, we use the same learning rate and weight-decay coefficients as entry #12.
- **Modification 1.** All diagnostic trajectories and training runs use a single H100 80GB GPU. Each run lasts 3300 steps and uses a learning-rate schedule with linear decay beginning after 30% of the training steps.
- **Modification 2.** Each optimizer step accumulates gradients over eight microbatches, giving an effective batch matching the batch size of entry #12.
- **Modification 3.** In the type-specific learning-rate setting, we use learning rates of 0.025 for W_Q, W_K, and W_V, 0.030 for W_O, and 0.035 for W_{up} and W_{down}. The weight-decay coefficient is 0.025 for all six matrix types except W_{up}, for which it is 0.0375. This type-specific configuration gives Muon a slightly lower validation loss than the shared-learning-rate configuration.

Diagnostic evaluation. To evaluate the local diagnostics developed in Section 3 at each checkpoint, we compute the quantities required by the local quadratic model. Specifically, for each Muon-updated block i, we compute the weight norm $w_i = \|W_i\|_F$, the training-loss gradient $G_i = \nabla_{W_i}\mathcal{L}$, and the corresponding unit-Frobenius Muon update direction V_i. These give the first-order descent coefficient $a_i = \langle G_i, V_i\rangle_F$. We further compute the directional Hessian interactions

$$C_{ij} = \langle V_i, H_{ij}[V_j]\rangle_F$$

using exact Hessian–vector products (HVPs), where $H_{ij}[V_j]$ is the directional derivative of G_i with respect to W_j along V_j. All gradients and HVPs are evaluated on the full training batch at each checkpoint. To fit memory, we split the full batch into 128 microbatches and accumulate the resulting gradients and HVPs.

A.2 Algorithmic Details

Practical implementation of MuonL and t-MuonL. For any alignment policy, we define the *alignment scale* of block i as $s_i := x_i/r_i$, where r_i is the u-w ratio before alignment and x_i is the ratio after alignment (see Equation 2). The Muon update U_i is therefore multiplied by s_i before scaling by the learning rate. In practical implementations, both MuonL and t-MuonL refresh their alignment scales every five optimizer steps and smooth them using an exponential moving average (EMA) with coefficient 0.9 and bias correction (Kingma & Ba, 2015). We denote the resulting EMA-smoothed, bias-corrected scale actually applied to U_i by $\hat{s}_i$. When computing the scales s_i, t-MuonL cancels the common ideal update norm $\sqrt{m_i}$ within each matrix type, as prescribed by Muon's shape scaling (Jordan et al., 2024b), thereby ignoring the deviations introduced by the finite-step Newton-Schulz approximation; MuonL, by contrast, computes its alignment scales using the actual Frobenius norms $\|U_i\|_F$ of the Muon updates for individual blocks. In contrast, the diagnostic evaluation directly uses the u-w ratios x_i given by the alignment policies, without the refresh or EMA smoothing.

Zero-initialization handling. Under the `modded-nanogpt` benchmark convention (Jordan, 2026), the projection matrices W_O and W_{down} are zero-initialized. Zero or near-zero weight norms can render the u-w ratios r_i undefined or ill-conditioned, thereby compromising the numerical stability of the alignment scales s_i. Therefore, we handle these cases using a safety threshold $\epsilon = 10^{-12}$:

- For MuonL, a block i with $\|W_i\|_F \le \epsilon$ is excluded from the global harmonic mean, and its alignment scale s_i is directly set to 1.
- For t-MuonL, when a matrix type $\mathcal{G}$'s average weight-norm $|\mathcal{G}|^{-1}\sum_{i\in\mathcal{G}}\|W_i\|_F \le \epsilon$, every block i in $\mathcal{G}$ is assigned an alignment scale $s_i = 1$.

Treatment of decoupled weight decay. All training runs without the u/w floor reported in Section 4 and Appendix A.3 use decoupled weight decay, with Muon, MuonL, and t-MuonL using the same weight-decay coefficients. Specifically, for each Muon-updated block i, the weight-decay term is applied separately from u-w ratio alignment as in

$$W_i \leftarrow (1 - \eta_i\lambda_i)W_i + \Delta_i(x),$$

where $\Delta_i(x)$ is the parameter change after alignment (see Equation 1), and η_i and λ_i denote the learning rate and weight-decay coefficient for block i, respectively. However, to isolate the effect of u-w ratio alignment on the Muon update, we omit the weight-decay term from the optimization potential scores and corresponding criteria, as specified in Section 2.

Given the Muon updates U_i, Algorithms 1 and 2 specify how MuonL and t-MuonL, respectively, compute the parameter changes $\Delta_i(x)$ after u-w ratio alignment. The algorithms incorporate the practical refresh and EMA scheme, as well as the zero-initialization handling described above.

Algorithm 1 MuonL

Require: weight matrices $\{W_i\}_{i\in\mathcal{I}}$ with $w_i := \|W_i\|_F$, pre-learning-rate Muon updates $\{U_i\}_{i\in\mathcal{I}}$, learning rates $\{\eta_i\}_{i\in\mathcal{I}}$, current optimizer step t
Hyperparameters: refresh interval K, EMA coefficient β, weight-norm threshold ϵ
State: $\hat{s}_i \leftarrow 1$, $e_i \leftarrow 0$ for all $i \in \mathcal{I}$, and $n \leftarrow 0$

if $t \bmod K = 0$ **then**
  $s_i \leftarrow 1$ for all $i \in \mathcal{I}$
  $\mathcal{I}_+ \leftarrow \{i \in \mathcal{I} : w_i > \epsilon\}$
  if $\mathcal{I}_+ \neq \varnothing$ **then**
    Compute $r_i \leftarrow \|U_i\|_F / w_i$ for all $i \in \mathcal{I}_+$ and $\bar{r} \leftarrow \left(|\mathcal{I}_+|^{-1} \sum_{i\in\mathcal{I}_+} r_i^{-1}\right)^{-1}$
    $s_i \leftarrow \bar{r}/r_i$ for all $i \in \mathcal{I}_+$
  end if
  $n \leftarrow n + 1$
  $e_i \leftarrow \beta e_i + (1-\beta)s_i$, $\hat{s}_i \leftarrow e_i/(1-\beta^n)$ for all $i \in \mathcal{I}$
end if
$\Delta_i \leftarrow -\eta_i \hat{s}_i U_i$ for all $i \in \mathcal{I}$
return $\{\Delta_i\}_{i\in\mathcal{I}}$

Algorithm 2 t-MuonL

Require: weight matrices $\{W_i\}_{i\in\mathcal{I}}$ with $w_i := \|W_i\|_F$, pre-learning-rate Muon updates $\{U_i\}_{i\in\mathcal{I}}$, learning rates $\{\eta_i\}_{i\in\mathcal{I}}$, current optimizer step t, matrix-type partition $\mathfrak{G}$
Hyperparameters: refresh interval K, EMA coefficient β, weight-norm threshold ϵ
State: $\hat{s}_i \leftarrow 1$, $e_i \leftarrow 0$ for all $i \in \mathcal{I}$, and $n \leftarrow 0$

if $t \bmod K = 0$ **then**
  $s_i \leftarrow 1$ for all $i \in \mathcal{I}$
  for $\mathcal{G} \in \mathfrak{G}$ **do**
    if $|\mathcal{G}|^{-1} \sum_{i\in\mathcal{G}} w_i > \epsilon$ **then**
      $s_i \leftarrow \dfrac{|\mathcal{G}| w_i}{\sum_{j\in\mathcal{G}} w_j}$ for all $i \in \mathcal{G}$
    end if
  end for
  $n \leftarrow n + 1$
  $e_i \leftarrow \beta e_i + (1-\beta)s_i$, $\hat{s}_i \leftarrow e_i/(1-\beta^n)$ for all $i \in \mathcal{I}$
end if
$\Delta_i \leftarrow -\eta_i \hat{s}_i U_i$ for all $i \in \mathcal{I}$
return $\{\Delta_i\}_{i\in\mathcal{I}}$

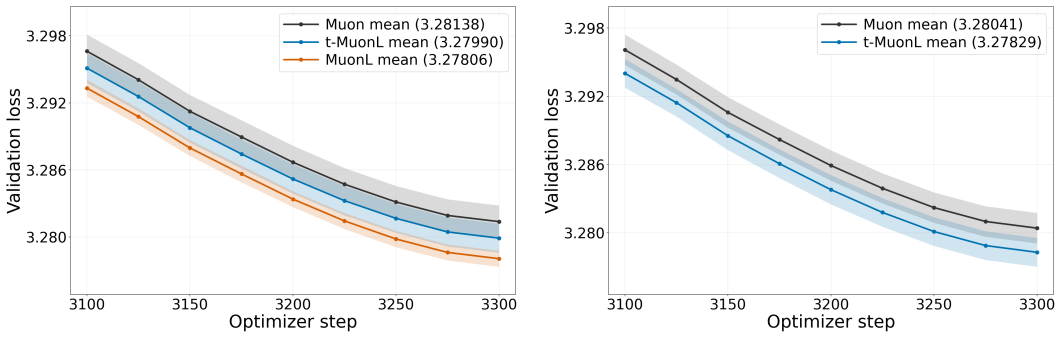
A.3 Additional Evidence

Validation loss across additional seeds. We repeat the training runs for Muon, MuonL, and t-MuonL with five additional random seeds under the same configurations as in Section 4. Figure 5(a) shows that, under a shared learning rate, both t-MuonL and MuonL achieve lower mean validation losses than Muon late in training. Figure 5(b) shows that, under type-specific learning rates, t-MuonL also achieves lower mean validation loss than Muon late in training. These results further corroborate the advantages of MuonL and t-MuonL over Muon indicated by the diagnostic criteria developed in Section 3.

Benchmark validation of MuonL. We next apply the statistical validation rule in the `modded-nanogpt` optimization benchmark (Track 3) to MuonL. The rule determines the earliest training step at which an optimizer reaches the target validation loss with sufficient evidence across n non-cherry-picked trials (Jordan, 2026). The benchmark considers a step count statistically validated when

$$(3.28 - \overline{L})\sqrt{n} \geq 0.004,$$

where $\overline{L}$ is the mean validation loss across the trials. Using the shared-learning-rate configuration adapted from benchmark entry #12 and described in Appendix A.1, we run 15 non-cherry-



(a) (b)

Figure 5: **Late-stage validation loss across five random seeds.** Curves show the mean validation loss; shaded bands denote one standard deviation across seeds. **(a)** Muon, t-MuonL, and MuonL under a shared learning rate. **(b)** t-MuonL versus Muon under type-specific learning rates.

picked MuonL trials. The earliest step satisfying this rule is 3260, where $\overline{L} = 3.27889$ and $(3.28 - \overline{L})\sqrt{15} = 0.00431$. Table 1 reports the mean validation loss and corresponding validation statistic near this crossing point. In contrast, entry #12 (the corresponding Muon baseline) first satisfies the same rule at step 3325. MuonL therefore achieves a statistically validated step count 65 steps earlier than Muon.

Table 1: **Benchmark validation of MuonL across 15 non-cherry-picked trials.** Here $\overline{L}$ is the mean validation loss across the trials. The validation rule is satisfied when $(3.28 - \overline{L})\sqrt{15} \geq 0.004$.

Validation step	$\overline{L}$	$(3.28 - \overline{L})\sqrt{15}$	≥ 0.004
3250	3.27943	0.00222	No
3260	**3.27889**	**0.00431**	**Yes**
3270	3.27844	0.00603	Yes
3280	3.27805	0.00755	Yes

Sanity check on diagnostic assumptions. We verify the curvature conditions required by the diagnostic criteria in Proposition 1 and Proposition 2 on the Muon trajectories from the preceding experiments. Figure 6(a) shows that, under a shared learning rate, the curvature terms $B(x^{\text{base}})$ for Muon and $B(x^{\text{full}})$ for MuonL in Proposition 1 are positive at all 13 checkpoints. Figure 6(b) shows that, under type-specific learning rates, the matrices $\mathbf{K}$ and $\mathbf{DKD} - \mathbf{\Gamma}$ in Proposition 2 are positive definite at all 13 checkpoints. The positive definiteness of both matrices in Figure 6(b) also implies the within-type curvature conditions in Proposition 1. Recall that $w_i, \langle w_i \rangle_{\mathcal{G}}$ respectively denote the weight norm of block i and the weight-norm average within type $\mathcal{G}$, and that C_{ij} measures the unit Hessian interaction between the updates of blocks i, j. Let $m_{\mathcal{G}}$ denote the common row dimension of matrices in $\mathcal{G}$. Substituting the u-w ratios $x_i^{\text{base}} = \sqrt{m_{\mathcal{G}}}/w_i$ for Muon and $x_i^{\text{typed}} = \sqrt{m_{\mathcal{G}}}/\langle w_i \rangle_{\mathcal{G}}$ for t-MuonL into the within-type curvature expression

$$B_{\mathcal{G}}(x) = \sum_{i,j \in \mathcal{G}} x_i x_j w_i w_j C_{ij},$$

and using Equation 13 and Equation 14, we have

$$B_{\mathcal{G}}(x^{\text{base}}) = m_{\mathcal{G}} |\mathcal{G}|^2 \mathbf{K}_{\mathcal{G},\mathcal{G}}, \quad B_{\mathcal{G}}(x^{\text{typed}}) = m_{\mathcal{G}} |\mathcal{G}|^2 (\mathbf{DKD} - \mathbf{\Gamma})_{\mathcal{G},\mathcal{G}}.$$

Both quantities are positive because positive definite matrices have positive diagonal entries.

Deviation from shape scaling. As discussed in Section 3.2, Muon's shape scaling determines the norm of a Muon update. Because u-w ratio alignment multiplies each Muon update by the alignment scale s_i, the alignment scale itself specifies the distortion of shape scaling. We quantify this distortion by the absolute percentage deviation (APD) of block i:

$$\text{APD}_i := 100|s_i - 1|\%. \tag{A.1}$$

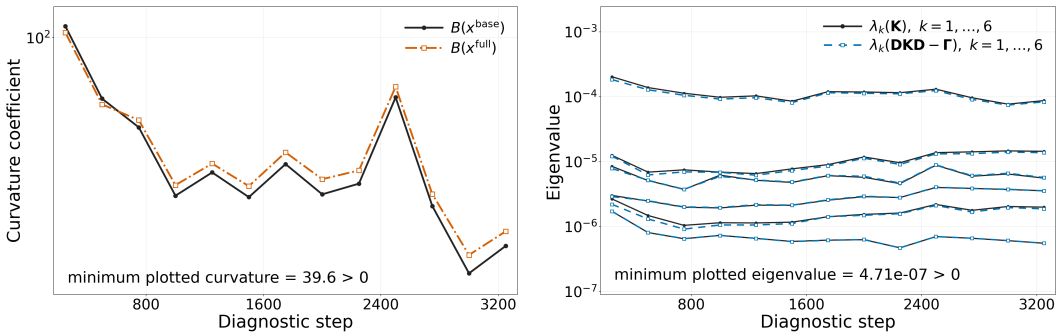


Figure 6: **Sanity checks for the diagnostic curvature assumptions. (a)** Curvature terms $B(x^{\text{base}})$ for Muon and $B(x^{\text{full}})$ for MuonL along the shared-learning-rate Muon trajectory. **(b)** Eigenvalues of the matrices $\mathbf{K}$ and $\mathbf{DKD} - \mathbf{\Gamma}$ in Proposition 2 along the type-specific learning-rate Muon trajectory. Both panels use logarithmic vertical scales.

Using Equation 2 to compute $s_i = x_i / r_i$, the alignment scales for t-MuonL (within type $\mathcal{G}$) and MuonL are, respectively,

$$\frac{\|W_i\|_F}{\text{mean}_{j \in \mathcal{G}(i)} \|W_j\|_F} \quad \text{and} \quad \frac{\|W_i\|_F / \sqrt{m_i}}{\text{mean}_{j \in \mathcal{I}} \left(\|W_j\|_F / \sqrt{m_j} \right)}. \tag{A.2}$$

Compared with t-MuonL, MuonL incorporates variation in m_j across types and averages over a larger set of blocks, which may produce alignment scales farther from one. Figure 7 reports the APDs induced by the two alignment policies along the Muon trajectory in the shared-learning-rate setting. For each diagnostic step and matrix type, it reports the median and maximum APDs across the 12 blocks of that type. Averaged over all diagnostic steps and matrix types, the median and maximum APDs are 5.26% and 12.88% for t-MuonL, versus 10.37% and 16.61% for MuonL. These results support t-MuonL as an alignment policy that causes less distortion of Muon's shape scaling than MuonL.

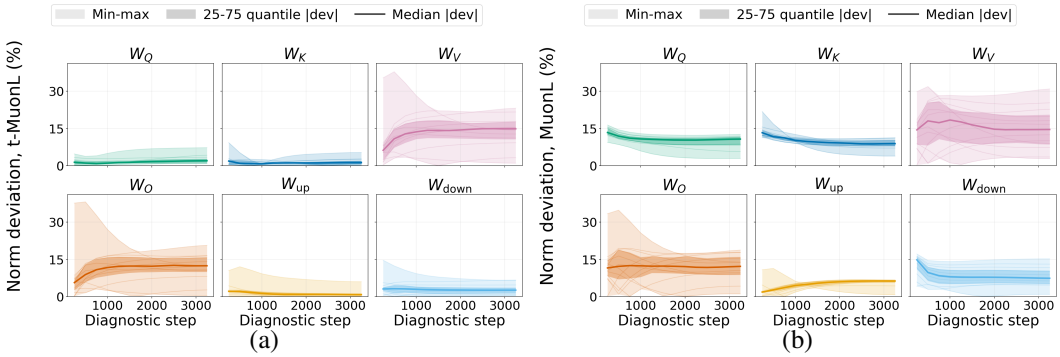


Figure 7: **Distortion of Muon's shape scaling by U-W ratio alignment.** The plots show the APDs of the alignment scales induced by t-MuonL and MuonL along the shared-learning-rate Muon trajectory. The six panels in each subfigure correspond to the six Muon-updated matrix types. Each panel tracks the 12 matrices of one type, one from each Transformer layer. Thin curves show individual blocks; the thick curve shows their median; and the dark and light bands show the 25th–75th percentile and min–max ranges, respectively. **(a)** shows t-MuonL and **(b)** MuonL.

B Comparison with OrScale

B.1 Algorithmic Differences

OrScale with the vision configuration (hereafter *OrScale-V*) aligns the *trust ratios* $\|W_i\|_F / \|U_i\|_F$ across Muon-updated blocks (Lou & You, 2026). Because the trust ratio is the inverse of the u-w

ratio, this is equivalent to aligning u-w ratios. The work also evaluates OrScale-V on CIFAR-10/DavidNet and derives measurable conditions under which its theoretical descent bound improves over Muon's. OrScale-V is therefore the closest prior work to our study of u-w ratio alignment. However, the algorithms of MuonL and OrScale-V differ in two respects: the alignment mechanism and the weight-decay coupling, as summarized in Table 2.

Table 2: Algorithmic comparison between MuonL and OrScale-V.

Aspect	**MuonL**	**OrScale-V**
Alignment mechanism	Aligns the u-w ratios to their harmonic mean across all Muon-updated blocks.	Rescales each block's update to have the same norm as its weight matrix.
Weight-decay coupling	Aligns the u-w ratios $\|U_i\|_F/\|W_i\|_F$ and applies decoupled weight decay as a separate step.	Aligns the quantity $\|W_i\|_F/\|\lambda W_i + U_i\|_F$, coupling the weight-decay term to the alignment process.

Specifically, in the quantity aligned by OrScale-V, λ is the weight-decay coefficient. It treats the weight-decay term as part of the update before alignment, with the aim of preventing runaway weight-norm growth. However, as we show in Section B.3, this coupling is unnecessary.

B.2 Contributions beyond OrScale

Beyond the algorithmic differences above, our work makes three additional contributions relative to OrScale-V: improved criterion accuracy, algorithmic originality through t-MuonL, and an expanded application scope.

Criterion accuracy. Using the notation from Lou & You (2026), let f denote the population loss and W_ℓ the weight matrix of layer ℓ. For each layer ℓ, OrScale-V assumes a smoothness constant L_ℓ such that whenever W and W' differ only in block ℓ, $\|\nabla_\ell f(W) - \nabla_\ell f(W')\|_F \le L_\ell \|W_\ell - W'_\ell\|_F$. Let y_ℓ denote the alignment scale for layer ℓ. OrScale-V evaluates alignment policies using the objective

$$\Phi(y) := \frac{\left(\sum_\ell y_\ell a_\ell\right)^2}{\sum_\ell L_\ell p_\ell y_\ell^2}, \qquad a_\ell := \|\nabla_\ell f\|_*, \quad p_\ell := \min(m_\ell, n_\ell),$$

where m_ℓ and n_ℓ are the row and column dimensions of W_ℓ, and $\|\cdot\|_*$ denotes the nuclear norm. Although the OrScale-V objective and our shared-learning-rate score $\mathcal{S}_{\mathrm{SL}}(x)$ have similar forms, they encode curvature differently in their denominators. The OrScale-V denominator incorporates assumed layerwise smoothness constants L_ℓ, rather than evaluated directional curvature. Moreover, it contains no cross-layer curvature interactions. By contrast, we evaluate the directional Hessian terms C_{ij} with HVPs and use them in our score, including the off-diagonal terms that capture cross-block interactions. This enables our criterion to more accurately characterize the local curvature relevant to u-w ratio alignment.

Introduction of t-MuonL. In contrast to OrScale-V, our work additionally introduces t-MuonL that is compatible with type-specific learning-rate settings. This broadens the potential applicability of u-w ratio alignment to LLM pretraining, where type-specific learning rates can be beneficial (Wang et al., 2025; Wen et al., 2026b). Rather than applying full alignment across all blocks, t-MuonL aligns u-w ratios only within each of the six Transformer matrix types $(W_Q, W_K, W_V, W_O, W_{\mathrm{up}}, W_{\mathrm{down}})$. Under a shared learning rate, t-MuonL also causes less distortion of Muon's shape scaling than MuonL, as assessed in Appendix A.3.

Application to LLM pretraining. OrScale-V has been evaluated on the CIFAR-10/DavidNet image classification task. However, the systematic LLM experiments in Lou & You (2026) use OrScale with the LM configuration (hereafter *OrScale-LM*), which fixes each block's trust ratio at the ratio recorded at that block's first nonzero update, rather than aligning trust ratios across blocks. Thus, neither the OrScale-V nor the OrScale-LM experiment establishes whether u-w ratio alignment

across blocks is helpful for LLM pretraining. In contrast, we directly analyze and implement u-w ratio alignment in the `modded-nanogpt` pretraining setting. The local diagnostics favor alignment in both learning-rate settings studied in Section 3, and the corresponding training runs achieve lower validation losses with t-MuonL or MuonL than with Muon.

B.3 Runaway Prevention

Lou & You (2026) identify a positive feedback loop under trust-ratio alignment with decoupled weight decay: a larger weight norm increases the trust ratio $\|W_i\|_F/\|U_i\|_F$, which amplifies the update and further increases the weight norm. This problem indeed exists; however, it arises from directly rescaling each layer's update norm to match its weight norm, not from decoupled weight decay itself. To isolate this mechanism, we define *decoupled-OrScale-V* as an ablation that excludes weight decay from the trust-ratio alignment process and applies it as a separate decoupled step.

To make this separation explicit, let t index the optimizer step. For each block i, the first and second lines below give the update rules for OrScale-V and decoupled-OrScale-V, respectively:

$$\begin{cases} \tau_{i,t}^{\mathrm{O}} := \dfrac{\|W_{i,t}\|_F}{\|\lambda W_{i,t} + U_{i,t}\|_F}, & \text{then} \quad W_{i,t+1} \leftarrow W_{i,t} - \eta\tau_{i,t}^{\mathrm{O}}(\lambda W_{i,t} + U_{i,t}), \\ \tau_{i,t}^{\mathrm{D}} := \dfrac{\|W_{i,t}\|_F}{\|U_{i,t}\|_F}, & \text{then} \quad W_{i,t+1} = (1-\eta\lambda)W_{i,t} - \eta\tau_{i,t}^{\mathrm{D}}U_{i,t}, \end{cases} \tag{B.1}$$

where $W_{i,t}$ and $U_{i,t}$ are the weight matrix and pre-learning-rate Muon update at step t, η is the learning rate, and λ is the weight-decay coefficient. In real implementations, OrScale-V clips the alignment scale $\tau_{i,t}$ to prescribed lower and upper bounds. However, we omit clipping here in Equation B.1 to directly analyze the feedback caused by the alignment mechanism.

Motivated by empirical observations in Muon-based LLM pretraining (Apte, 2026), Proposition 3 below assumes a persistent outward radial component γ in the Muon update. Under the stated assumptions, MuonL[7] keeps the weight norms bounded, whereas unclipped decoupled-OrScale-V yields exponentially growing weight norms when γ is sufficiently large.

Proposition 3 (Weight-norm runaway prevention of MuonL) *Consider MuonL and the decoupled-OrScale-V (without clipping) update in Equation B.1, using the same constant learning rate $\eta > 0$ and weight-decay coefficient $\lambda > 0$, with $0 < \eta\lambda < 1$. Let $V_{i,t} := U_{i,t}/\|U_{i,t}\|_F$ denote the unit-Frobenius Muon update direction. For both methods, assume that $\|W_{i,0}\|_F > 0$ for every block $i \in \mathcal{I}$, and that the Muon update has a persistent outward radial component given by a constant $\gamma \in (0, 1]$:*

$$\left\langle -V_{i,t}, \frac{W_{i,t}}{\|W_{i,t}\|_F} \right\rangle_F = \gamma \qquad \text{for every } i \in \mathcal{I} \text{ and } t \geq 0.$$

Then, under MuonL, each weight norm $\|W_{i,t}\|_F$ converges to a finite positive limit. By contrast, under unclipped decoupled-OrScale-V, each weight norm grows exponentially whenever

$$\gamma > \frac{2\lambda - \eta(1+\lambda^2)}{2(1-\eta\lambda)}.$$

For small η, this condition is approximately $\gamma > \lambda$.

Thus, the alignment mechanism in our work can prevent weight-norm runaway without coupling the weight-decay term into the alignment process. This conclusion is also supported by the weight-norm dynamics observed in training. Figure 8 reports these dynamics for MuonL and the clipped versions[8] of OrScale-V and decoupled-OrScale-V. Apart from the update rule and the treatment of the weight-decay term, all three runs use the same training configuration. These clipped runs do not directly test the exponential growth predicted for unclipped decoupled-OrScale-V in Proposition 3. Nevertheless, we can observe that decoupled-OrScale-V (with clipping) develops substantially larger weight norms than MuonL and OrScale-V (with clipping) across all six matrix types, and MuonL controls weight norms as effectively as OrScale-V (with clipping).

[7] Throughout this section, we consider the idealized MuonL update obtained by applying x^{full} in Equation 1, without the EMA smoothing or periodic refresh used in practice.

[8] For both OrScale-V and decoupled-OrScale-V, we restrict each block's alignment scale $\tau_{i,t}$ to $[0.1, 5.0]$, following the LLM-experiment setting of Lou & You (2026).

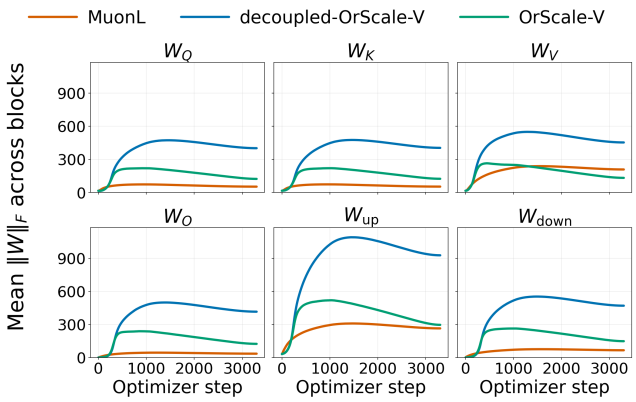


Figure 8: **Weight-norm dynamics under MuonL, OrScale-V, and decoupled-OrScale-V.** Each panel corresponds to one Transformer matrix type and shows the mean Frobenius weight norm across its 12 blocks over 3300 training steps. Both OrScale-V variants clip their alignment scales to $[0.1, 5.0]$.

C Additional Theory for U-W Ratio Alignment

In Appendix C.1, we provide the derivations of Equations 12 and 16 used in Section 3.3. In Appendices C.2 and C.3, we extend the criteria for when u-w ratio alignment benefits Muon in Propositions 1 and 2 to general optimizers.

C.1 Derivations for the Type-Specific Learning-Rate Analysis

Derivation of the type-specific score. We first derive the score in Equation 12, which measures the optimization potential of an alignment policy x under type-specific learning rates. Recall the first-order vector $\mathbf{A}(x) \in \mathbb{R}^N$, curvature matrix $\mathbf{B}(x) \in \mathbb{R}^{N\times N}$, and type-specific learning-rate vector $\boldsymbol{\eta} = (\eta_{\mathcal{G}})_{\mathcal{G}\in\mathfrak{G}} \in \mathbb{R}^N$ from Section 3.3, where N is the number of matrix types and the entries of $\mathbf{A}(x), \mathbf{B}(x)$ are

$$A_{\mathcal{G}}(x) = \sum_{i\in\mathcal{G}} x_i w_i a_i, \qquad B_{\mathcal{G},\mathcal{H}}(x) = \sum_{i\in\mathcal{G},\, j\in\mathcal{H}} x_i x_j w_i w_j C_{ij}.$$

Here a_i is the first-order descent coefficient and C_{ij} is the directional Hessian interaction between blocks i and j. For a block $i \in \mathcal{G}$, we write the parameter update according to Equation 1, giving

$$\Delta_i(x) = -\eta_{\mathcal{G}} x_i w_i V_i, \tag{C.1}$$

where $w_i = \|W_i\|_F$ is the weight norm and V_i is the unit-Frobenius update direction. Because all blocks in a fixed type $\mathcal{G}$ share the learning rate $\eta_{\mathcal{G}}$, their first-order contributions combine as

$$\eta_{\mathcal{G}} \sum_{i\in\mathcal{G}} x_i w_i a_i = \eta_{\mathcal{G}} A_{\mathcal{G}}(x). \tag{C.2}$$

Likewise, the contributions of block pairs in $\mathcal{G}\times\mathcal{H}$ combine as

$$\eta_{\mathcal{G}}\eta_{\mathcal{H}} \sum_{i\in\mathcal{G},\, j\in\mathcal{H}} x_i x_j w_i w_j C_{ij} = \eta_{\mathcal{G}}\eta_{\mathcal{H}} B_{\mathcal{G},\mathcal{H}}(x). \tag{C.3}$$

Substituting the parameter changes from Equation C.1 into the Taylor expansion in Equation 3, and then collecting the first-order and curvature terms as in Equations C.2 and C.3, yields

$$\mathcal{L}(W) - \mathcal{L}(W + \Delta(x)) = \boldsymbol{\eta}^\top \mathbf{A}(x) - \frac{1}{2}\boldsymbol{\eta}^\top \mathbf{B}(x)\boldsymbol{\eta} + o(\|\Delta(x)\|_F^2). \tag{C.4}$$

Since we assume $\mathbf{B}(x)$ to be positive definite, differentiating the quadratic part with respect to $\boldsymbol{\eta}$ gives $\mathbf{A}(x) - \mathbf{B}(x)\boldsymbol{\eta} = 0$. Then the unique maximizer is

$$\boldsymbol{\eta}^\star(x) = \mathbf{B}(x)^{-1}\mathbf{A}(x).$$

Substituting it into Equation C.4 yields the type-specific score in Equation 12:

$$\mathcal{S}_{\mathrm{TL}}(x) := \frac{1}{2}\mathbf{A}(x)^\top \mathbf{B}(x)^{-1}\mathbf{A}(x), \tag{C.5}$$

which measures the optimized potential given by policy x when all matrix types are updated together.

Derivation of the generalized-eigenvalue expression. We next derive Equation 16, which writes the generalized eigenvalue ρ_k in Proposition 2 as a sum involving covariance terms $c_{\mathcal{G}}$ and $q_{\mathcal{G},\mathcal{H}}$, where

$$c_{\mathcal{G}} = \mathrm{cov}_{\mathcal{G}}\left([w_i]_{\mathcal{G}}, [a_i]_{\mathcal{G}}\right), \quad q_{\mathcal{G},\mathcal{H}} = \mathrm{cov}_{\mathcal{G}\times\mathcal{H}}\big([w_i]_{\mathcal{G}}[w_j]_{\mathcal{H}}, C_{ij}\big).$$

Recall Equation 14, which defines the symmetric matrix $\mathbf{\Gamma}$ by specifying each entry

$$\mathbf{\Gamma}_{\mathcal{G},\mathcal{H}} = (c_{\mathcal{G}} + c_{\mathcal{H}} + c_{\mathcal{G}}c_{\mathcal{H}})K_{\mathcal{G},\mathcal{H}} - q_{\mathcal{G},\mathcal{H}}$$

and also recall that $\mathbf{D}$ is diagonal with entries $1 + c_{\mathcal{G}}$. Substituting the expression for $\mathbf{\Gamma}_{\mathcal{G},\mathcal{H}}$ gives

$$\begin{aligned}(\mathbf{D}^{-1}\mathbf{\Gamma}\mathbf{D}^{-1})_{\mathcal{G},\mathcal{H}} &= \frac{\mathbf{\Gamma}_{\mathcal{G},\mathcal{H}}}{(1+c_{\mathcal{G}})(1+c_{\mathcal{H}})} \\ &= K_{\mathcal{G},\mathcal{H}} - \frac{K_{\mathcal{G},\mathcal{H}} + q_{\mathcal{G},\mathcal{H}}}{(1+c_{\mathcal{G}})(1+c_{\mathcal{H}})}.\end{aligned} \tag{C.6}$$

For the generalized eigenpair $(\rho_k, \mathbf{v}_k)$ in Proposition 2, left-multiplying $\mathbf{D}^{-1}\mathbf{\Gamma}\mathbf{D}^{-1}\mathbf{v}_k = \rho_k\mathbf{K}\mathbf{v}_k$ by $\mathbf{v}_k^\top$ and using $\mathbf{v}_k^\top\mathbf{K}\mathbf{v}_k = 1$ gives $\rho_k = \mathbf{v}_k^\top\mathbf{D}^{-1}\mathbf{\Gamma}\mathbf{D}^{-1}\mathbf{v}_k$. Substituting Equation C.6 and expanding over all matrix-type pairs yields

$$\rho_k = \sum_{\mathcal{G},\mathcal{H}} v_{k,\mathcal{G}}v_{k,\mathcal{H}}\left[K_{\mathcal{G},\mathcal{H}} - \frac{K_{\mathcal{G},\mathcal{H}} + q_{\mathcal{G},\mathcal{H}}}{(1+c_{\mathcal{G}})(1+c_{\mathcal{H}})}\right], \tag{C.7}$$

which is Equation 16.

C.2 Shared-Learning-Rate Criteria for General Optimizers

We now extend the sufficient criteria in Proposition 1 to general optimizers. We give conditions for within-type alignment to improve the within-type score $\mathcal{S}_{\mathcal{G}}$ and for full alignment to improve the shared-learning-rate score $\mathcal{S}_{\mathrm{SL}}$, each relative to the original optimizer.

Recall the weight norm $w_i = \|W_i\|_F$, the original u-w ratio $r_i = \|U_i\|_F/w_i$, and the first-order descent coefficient a_i along the unit-Frobenius update direction $-V_i$. The adjusted directional curvatures $\widetilde{L}_i^{\mathcal{G}}$ and $\widetilde{L}_i^{\mathcal{I}}$ include block i's curvature and its interactions with other blocks in $\mathcal{G}$ and $\mathcal{I}$, respectively. Recall from Definition 1 that $\langle f\rangle_\pi = \sum_i \pi_i f_i$ is the weighted mean and $[f_i]_\pi = f_i/\langle f\rangle_\pi$ is the weighted normalization whenever $\langle f\rangle_\pi \neq 0$. The corresponding weighted covariance and variance are denoted by cov_π and var_π, respectively.

Proposition 4 (Sufficient criteria for u-w ratio alignment) *Let* $\pi_i = (r_iw_i)^2/\sum_{j\in\mathcal{G}}(r_jw_j)^2$. *Assume the within-type curvature terms* $B_{\mathcal{G}}(x^{\mathrm{typed}}), B_{\mathcal{G}}(x^{\mathrm{base}})$ *are positive with first-order term* $A_{\mathcal{G}}(x^{\mathrm{base}}) \neq 0$. *If*

$$2\,\mathrm{cov}_\pi\left([r_i^{-1}]_\pi, [a_i/(r_iw_i)]_\pi\right) - \mathrm{cov}_\pi\left(([r_i^{-1}]_\pi)^2, [\widetilde{L}_i^{\mathcal{G}}]_\pi\right) > \mathrm{var}_\pi([r_i^{-1}]_\pi), \tag{C.8}$$

then $\mathcal{S}_{\mathcal{G}}(x^{\mathrm{typed}}) > \mathcal{S}_{\mathcal{G}}(x^{\mathrm{base}})$*, i.e., the within-type alignment policy* x^{typed} *benefits the original optimizer in optimization potential within type* $\mathcal{G}$. *Let* $\nu_i = (r_iw_i)^2/\sum_{j\in\mathcal{I}}(r_jw_j)^2$. *Assume the curvature terms* $B(x^{\mathrm{full}}), B(x^{\mathrm{base}})$ *are positive with first-order term* $A(x^{\mathrm{base}}) \neq 0$. *If*

$$2\,\mathrm{cov}_\nu\left([r_i^{-1}]_\nu, [a_i/(r_iw_i)]_\nu\right) - \mathrm{cov}_\nu\left(([r_i^{-1}]_\nu)^2, [\widetilde{L}_i^{\mathcal{I}}]_\nu\right) > \mathrm{var}_\nu([r_i^{-1}]_\nu), \tag{C.9}$$

then $\mathcal{S}_{\mathrm{SL}}(x^{\mathrm{full}}) > \mathcal{S}_{\mathrm{SL}}(x^{\mathrm{base}})$*, i.e., the full alignment policy* x^{full} *benefits the original optimizer in optimization potential under a shared learning rate.*

As in Proposition 1, we refer to the three terms in Equation C.8, from left to right, as the first-order descent gain, curvature gain, and rescaling cost. Assuming all first-order entries a_i to be positive, the within-type criterion favors u-w alignment when blocks in $\mathcal{G}$ with smaller original u-w ratios exhibit larger values of $a_i/(r_iw_i)$ and lower adjusted directional curvatures, provided these gains outweigh the rescaling cost. The full-alignment criterion in Equation C.9 has the same interpretation, with weights and adjusted directional curvatures computed over all blocks in $\mathcal{I}$. It can be verified that Proposition 4 recovers both criteria for Muon in Proposition 1.

C.3 Type-Specific Learning-Rate Criterion for General Optimizers

We now extend the criterion in Proposition 2 to general optimizers. We compare within-type alignment with the original optimizer using the type-specific score $\mathcal{S}_{\mathrm{TL}}$, with each matrix type $\mathcal{G}$ using a type-specific learning rate $\eta_{\mathcal{G}}$.

Recall the original u-w ratios $r_i = \|U_i\|_F / w_i$, where $w_i = \|W_i\|_F$ is the weight norm. For each block $i \in \mathcal{G}$, the original update uses $x_i^{\mathrm{base}} = r_i$, while within-type alignment uses $x_i^{\mathrm{typed}} = \bar{r}_{\mathcal{G}}$, the harmonic mean of the original ratios in $\mathcal{G}$. Define probability weights proportional to the original update norms to be

$$\pi_i^{\mathcal{G}} := \frac{r_i w_i}{\sum_{j \in \mathcal{G}} r_j w_j}. \tag{C.10}$$

Recall the first-order descent coefficients a_i and directional Hessian interactions C_{ij} between blocks i and j. Using the weights in Equation C.10, we generalize Equation 13 by defining notation

$$\begin{aligned} g_{\mathcal{G}} &:= \langle a_i \rangle_{\pi^{\mathcal{G}}}, & K_{\mathcal{G},\mathcal{H}} &:= \langle C_{ij} \rangle_{\pi^{\mathcal{G}} \times \pi^{\mathcal{H}}}, \\ c_{\mathcal{G}} &:= \mathrm{cov}_{\pi^{\mathcal{G}}} \left([r_i^{-1}]_{\pi^{\mathcal{G}}}, [a_i]_{\pi^{\mathcal{G}}} \right), & \mathbf{D} &:= \mathrm{diag}\left(1 + c_{\mathcal{G}_1}, \ldots, 1 + c_{\mathcal{G}_N}\right). \end{aligned} \tag{C.11}$$

Here N is the number of matrix types. The average over $\pi^{\mathcal{G}} \times \pi^{\mathcal{H}}$ uses weights $\pi_i^{\mathcal{G}} \pi_j^{\mathcal{H}}$ for all $i \in \mathcal{G}$ and $j \in \mathcal{H}$, including $i = j$ when $\mathcal{G} = \mathcal{H}$. Write $\mathbf{g} = (g_{\mathcal{G}})_{\mathcal{G} \in \mathfrak{G}}$ and $\mathbf{K} = (K_{\mathcal{G},\mathcal{H}})_{\mathcal{G},\mathcal{H} \in \mathfrak{G}}$. We further generalize Equation 14 by defining the symmetric matrix $\mathbf{\Gamma}$ with entries

$$\begin{aligned} \Gamma_{\mathcal{G},\mathcal{H}} := \; & (c_{\mathcal{G}} + c_{\mathcal{H}} + c_{\mathcal{G}} c_{\mathcal{H}}) \, K_{\mathcal{G},\mathcal{H}} \\ & - \mathrm{cov}_{\pi^{\mathcal{G}} \times \pi^{\mathcal{H}}} \left([r_i^{-1}]_{\pi^{\mathcal{G}}} [r_j^{-1}]_{\pi^{\mathcal{H}}}, C_{ij} \right). \end{aligned} \tag{C.12}$$

Using these abstractions, we can derive the following criterion for whether within-type alignment results in higher optimization potential than the original optimizer under type-specific learning rates.

Proposition 5 (Criterion for within-type alignment with type-specific learning rates) *Assume the curvature matrices $\mathbf{K}$ and $\mathbf{DKD} - \mathbf{\Gamma}$ are positive definite, with $g_{\mathcal{G}} \neq 0$ and $c_{\mathcal{G}} \neq -1$ for every matrix type $\mathcal{G}$. Let $\{(\rho_k, \mathbf{v}_k)\}_{k=1}^N$ be the generalized eigenpairs of $(\mathbf{D}^{-1}\mathbf{\Gamma}\mathbf{D}^{-1}, \mathbf{K})$, with $\rho_k \in \mathbb{R}$ and $\mathbf{v}_k \in \mathbb{R}^N$ satisfying*

$$\mathbf{D}^{-1}\mathbf{\Gamma}\mathbf{D}^{-1}\mathbf{v}_k = \rho_k \mathbf{K}\mathbf{v}_k, \qquad \mathbf{v}_k^\top \mathbf{K} \mathbf{v}_\ell = \mathbf{1}\{k = \ell\}.$$

Then

$$\mathcal{S}_{\mathrm{TL}}(x^{\mathrm{typed}}) > \mathcal{S}_{\mathrm{TL}}(x^{\mathrm{base}}) \quad \Longleftrightarrow \quad \sum_{k=1}^{N} \frac{\rho_k}{1 - \rho_k} \left(\mathbf{v}_k^\top \mathbf{g} \right)^2 > 0. \tag{C.13}$$

Proposition 5 has exactly the same form as Proposition 2, except that the involved quantities $\mathbf{g}, \mathbf{K}, \mathbf{D}, \mathbf{\Gamma}$ are defined differently. Its interpretation also follows that of Proposition 2 accordingly.

D Theory Extension to SGD

As a basic example of how our framework extends to optimizers other than Muon, we apply the sufficient criteria in Proposition 4 to SGD (without momentum). In this section, we give conditions for within-type alignment to improve the within-type score $\mathcal{S}_{\mathcal{G}}$ and for full alignment to improve the shared-learning-rate score $\mathcal{S}_{\mathrm{SL}}$, each relative to SGD.

Recall that $w_i = \|W_i\|_F$ and $G_i = \nabla_{W_i} \mathcal{L}(W)$. For SGD without momentum, the pre-learning-rate update is $U_i = G_i$. The unit-Frobenius update direction V_i, first-order descent coefficient a_i, and original u-w ratio r_i therefore satisfy

$$V_i = \frac{G_i}{\|G_i\|_F}, \qquad a_i = \|G_i\|_F, \qquad r_i = \frac{\|G_i\|_F}{w_i}.$$

The first-order terms of SGD are naturally positive because $r_i w_i a_i = \|G_i\|_F^2 > 0$. The adjusted directional curvatures from Section 3.1 can now be written as

$$\widetilde{L}_i^{\mathcal{G}} = C_{ii} + \sum_{\substack{j \in \mathcal{G} \\ j \neq i}} \frac{2w_j \|G_j\|_F}{w_j \|G_i\|_F + w_i \|G_j\|_F} C_{ij},$$

$$\widetilde{L}_i^{\mathcal{I}} = C_{ii} + \sum_{\substack{j \in \mathcal{I} \\ j \neq i}} \frac{2w_j \|G_j\|_F}{w_j \|G_i\|_F + w_i \|G_j\|_F} C_{ij},$$

where C_{ij} is the directional Hessian interaction between blocks i and j. Substituting the SGD identities into Proposition 4 gives the following criteria. We reuse Definition 1 for the weighted mean, normalization, and covariance.

Corollary 1 (Sufficient criteria for SGD) *Let $\pi_i = \|G_i\|_F^2 / \sum_{j \in \mathcal{G}} \|G_j\|_F^2$. Assume the within-type curvature terms $B_{\mathcal{G}}(x^{\mathrm{typed}})$ and $B_{\mathcal{G}}(x^{\mathrm{base}})$ are positive. If*

$$-\operatorname{cov}_\pi \left(\left([w_i / \|G_i\|_F]_\pi \right)^2, [\widetilde{L}_i^{\mathcal{G}}]_\pi \right) > \operatorname{var}_\pi([w_i / \|G_i\|_F]_\pi), \tag{D.1}$$

then $\mathcal{S}_{\mathcal{G}}(x^{\mathrm{typed}}) > \mathcal{S}_{\mathcal{G}}(x^{\mathrm{base}})$, i.e., the within-type alignment policy x^{typed} benefits SGD in optimization potential within type $\mathcal{G}$. Let $\nu_i = \|G_i\|_F^2 / \sum_{j \in \mathcal{I}} \|G_j\|_F^2$. Assume the curvature terms $B(x^{\mathrm{full}})$ and $B(x^{\mathrm{base}})$ are positive. If

$$-\operatorname{cov}_\nu \left(\left([w_i / \|G_i\|_F]_\nu \right)^2, [\widetilde{L}_i^{\mathcal{I}}]_\nu \right) > \operatorname{var}_\nu([w_i / \|G_i\|_F]_\nu), \tag{D.2}$$

then $\mathcal{S}_{\mathrm{SL}}(x^{\mathrm{full}}) > \mathcal{S}_{\mathrm{SL}}(x^{\mathrm{base}})$, i.e., the full alignment policy x^{full} benefits SGD in optimization potential under a shared learning rate.

For SGD, $a_i/(r_i w_i) = 1$, so the first-order descent gain in Proposition 4 vanishes. The left- and right-hand sides of Equation D.1 are the curvature gain and rescaling cost, respectively. The within-type criterion favors u-w alignment when blocks in $\mathcal{G}$ with smaller original u-w ratios exhibit lower adjusted directional curvatures, provided that the curvature gain outweighs the rescaling cost. The full-alignment criterion has the same interpretation, with weights and adjusted directional curvatures computed over all SGD-updated blocks $\mathcal{I}$.

E Proofs

E.1 Proof of Proposition 1

We apply the general criteria in Proposition 4 to Muon. Recall that block i has dimensions $m_i \times n_i$, $k_i = \min\{m_i, n_i\}$, and $U_i = \sqrt{\max\{1, m_i/n_i\}}\, \operatorname{msgn}(M_i)$. The assumed full rank of M_i gives $\|\operatorname{msgn}(M_i)\|_F^2 = k_i$. Thus Muon's shape scaling implies

$$\|U_i\|_F^2 = \max\{1, m_i/n_i\}\, k_i = m_i, \qquad r_i w_i = \sqrt{m_i}, \qquad r_i^{-1} = \frac{w_i}{\sqrt{m_i}}. \tag{E.1}$$

Within a matrix type $\mathcal{G}$, let $m_{\mathcal{G}}$ denote the common row dimension. The within-type weights in the general criterion satisfy $\pi_i = m_{\mathcal{G}}/(|\mathcal{G}| m_{\mathcal{G}}) = 1/|\mathcal{G}|$. The normalized factors therefore reduce to

$$[r_i^{-1}]_\pi = [w_i/\sqrt{m_{\mathcal{G}}}]_{\mathcal{G}} = [w_i]_{\mathcal{G}},$$
$$[a_i/(r_i w_i)]_\pi = [a_i/\sqrt{m_{\mathcal{G}}}]_{\mathcal{G}} = [a_i]_{\mathcal{G}}.$$

Since the weights are uniform, the weighted covariance, variance, and normalization of the adjusted curvatures also become their unweighted counterparts within $\mathcal{G}$. Substituting into the general sufficient condition gives

$$2 \operatorname{cov}_{\mathcal{G}}([w_i]_{\mathcal{G}}, [a_i]_{\mathcal{G}}) - \operatorname{cov}_{\mathcal{G}}\left(([w_i]_{\mathcal{G}})^2, [\widetilde{L}_i^{\mathcal{G}}]_{\mathcal{G}} \right) > \operatorname{var}_{\mathcal{G}}([w_i]_{\mathcal{G}}).$$

This is precisely the stated within-type condition. The curvature and first-order assumptions are unchanged, so Proposition 4 gives $\mathcal{S}_{\mathcal{G}}(x^{\mathrm{typed}}) > \mathcal{S}_{\mathcal{G}}(x^{\mathrm{base}})$.

For full alignment, the general criterion uses weights

$$\nu_i = \frac{(r_i w_i)^2}{\sum_{j\in\mathcal{I}}(r_j w_j)^2} = \frac{m_i}{\sum_{j\in\mathcal{I}} m_j} = \pi_i,$$

where π_i denotes the weights in the full-alignment part of Proposition 1. Unlike the within-type comparison, m_i can vary across blocks, so the factors $\sqrt{m_i}$ remain inside the normalization:

$$[r_i^{-1}]_\nu = \left[\frac{w_i}{\sqrt{m_i}}\right]_\pi, \qquad [a_i/(r_i w_i)]_\nu = \left[\frac{a_i}{\sqrt{m_i}}\right]_\pi.$$

Substitution into the general full-alignment criterion therefore gives

$$\begin{aligned} &2\,\mathrm{cov}_\pi\left(\left[\frac{w_i}{\sqrt{m_i}}\right]_\pi, \left[\frac{a_i}{\sqrt{m_i}}\right]_\pi\right) \\ &\quad - \mathrm{cov}_\pi\left(\left(\left[\frac{w_i}{\sqrt{m_i}}\right]_\pi\right)^2, [\widetilde{L}_i^{\mathcal{I}}]_\pi\right) > \mathrm{var}_\pi\left(\left[\frac{w_i}{\sqrt{m_i}}\right]_\pi\right). \end{aligned}$$

This is the stated condition for MuonL outperforming Muon. Under the corresponding curvature and first-order assumptions, Proposition 4 yields $\mathcal{S}_{\mathrm{SL}}(x^{\mathrm{full}}) > \mathcal{S}_{\mathrm{SL}}(x^{\mathrm{base}})$. □

E.2 Proof of Proposition 2

We apply the general criterion in Proposition 5. For Muon, $r_i w_i = \sqrt{m_{\mathcal{G}}}$ for $i \in \mathcal{G}$, where $m_{\mathcal{G}}$ is the common row dimension within type $\mathcal{G}$. Thus the probability weights and normalized inverse u-w ratios satisfy

$$\pi_i^{\mathcal{G}} = \frac{\sqrt{m_{\mathcal{G}}}}{|\mathcal{G}|\sqrt{m_{\mathcal{G}}}} = \frac{1}{|\mathcal{G}|}, \qquad [r_i^{-1}]_{\pi^{\mathcal{G}}} = \frac{w_i/\sqrt{m_{\mathcal{G}}}}{\langle w\rangle_{\mathcal{G}}/\sqrt{m_{\mathcal{G}}}} = [w_i]_{\mathcal{G}}.$$

The product weights are also uniform over $\mathcal{G}\times\mathcal{H}$. Consequently, the first-order means, mean Hessian interactions, and within-type covariances become

$$\begin{aligned} g_{\mathcal{G}} &= \langle a_i\rangle_{\pi^{\mathcal{G}}} = \langle a_i\rangle_{\mathcal{G}}, \\ K_{\mathcal{G},\mathcal{H}} &= \langle C_{ij}\rangle_{\pi^{\mathcal{G}}\times\pi^{\mathcal{H}}} = \langle C_{ij}\rangle_{\mathcal{G}\times\mathcal{H}}, \\ c_{\mathcal{G}} &= \mathrm{cov}_{\pi^{\mathcal{G}}}\left([r_i^{-1}]_{\pi^{\mathcal{G}}}, [a_i]_{\pi^{\mathcal{G}}}\right) = \mathrm{cov}_{\mathcal{G}}\left([w_i]_{\mathcal{G}}, [a_i]_{\mathcal{G}}\right). \end{aligned}$$

Substituting the same identities into the pairwise covariance defining $\mathbf{\Gamma}$ gives

$$\begin{aligned} \Gamma_{\mathcal{G},\mathcal{H}} = (c_{\mathcal{G}} + c_{\mathcal{H}} + c_{\mathcal{G}}c_{\mathcal{H}})K_{\mathcal{G},\mathcal{H}} \\ - \mathrm{cov}_{\mathcal{G}\times\mathcal{H}}\left([w_i]_{\mathcal{G}}[w_j]_{\mathcal{H}}, C_{ij}\right). \end{aligned}$$

These are precisely the quantities $\mathbf{g}$, $\mathbf{K}$, and $\mathbf{\Gamma}$ in Proposition 2. Since the covariances $c_{\mathcal{G}}$ coincide, so does $\mathbf{D} = \mathrm{diag}(1 + c_{\mathcal{G}_1}, \ldots, 1 + c_{\mathcal{G}_N})$. The curvature assumptions and generalized eigenvalue problem are therefore unchanged. Proposition 5 then gives the stated comparison between t-MuonL and Muon. □

E.3 Proof of Proposition 3

Set $w_{i,t} := \|W_{i,t}\|_F$ and $\alpha := 1 - \eta\lambda \in (0,1)$. By assumption, $\langle W_{i,t}, V_{i,t}\rangle_F = -\gamma w_{i,t}$, where $\|V_{i,t}\|_F = 1$ and the same $\gamma \in (0,1]$ applies to every block and step. We first analyze MuonL's alignment mechanism from Equation 1, with decoupled weight decay.

By Equation E.1, the pre-learning-rate Muon update has norm $\sqrt{m_i}$ at every step. The harmonic-mean ratio assigned to all blocks is therefore

$$\bar{r}_t = \left(\frac{1}{|\mathcal{I}|}\sum_{j\in\mathcal{I}}\frac{w_{j,t}}{\sqrt{m_j}}\right)^{-1}.$$

The parameter update is $W_{i,t+1} = \alpha W_{i,t} - \eta\bar{r}_t w_{i,t} V_{i,t}$. Taking its squared Frobenius norm and substituting the radial-component assumption yields

$$\begin{aligned} w_{i,t+1}^2 &= \alpha^2 w_{i,t}^2 - 2\eta\alpha\bar{r}_t w_{i,t}\langle W_{i,t}, V_{i,t}\rangle_F + \eta^2\bar{r}_t^2 w_{i,t}^2 \\ &= \left(\alpha^2 + 2\eta\alpha\gamma\bar{r}_t + \eta^2\bar{r}_t^2\right) w_{i,t}^2. \end{aligned} \tag{E.2}$$

All norms remain positive, and the multiplicative factor is common to all blocks. Hence $w_{i,t} = c_t w_{i,0}$ for a common scalar $c_t > 0$ with $c_0 = 1$. The harmonic-mean definition then gives $\bar{r}_t = \bar{r}_0/c_t$. Dividing Equation E.2 by $w_{i,0}^2$ gives the scalar recurrence

$$c_{t+1} = \sqrt{\alpha^2 c_t^2 + 2\eta\alpha\gamma\bar{r}_0 c_t + \eta^2\bar{r}_0^2}. \tag{E.3}$$

To prove convergence, first observe that a positive fixed point must satisfy

$$(1-\alpha^2)c^2 - 2\eta\alpha\gamma\bar{r}_0 c - \eta^2\bar{r}_0^2 = 0.$$

The leading coefficient is positive and the constant term is negative, so this quadratic has exactly one positive root, denoted by $c_\star$. Subtracting c_t^2 from the square of Equation E.3 shows that $c_{t+1} > c_t$ when $0 < c_t < c_\star$, and $c_{t+1} < c_t$ when $c_t > c_\star$. The recurrence map is strictly increasing on $(0, \infty)$ because all coefficients inside its square root are positive. Since it fixes $c_\star$, an iterate below $c_\star$ remains below it, and an iterate above $c_\star$ remains above it. Thus (c_t) is monotone and bounded between 1 and $c_\star$. Its limit is positive and, by continuity of the recurrence map, must equal the unique positive fixed point $c_\star$. Consequently, $\lim_{t\to\infty} w_{i,t} = c_\star w_{i,0} \in (0, \infty)$ for every block i.

For decoupled-OrScale-V (without clipping), Equation B.1 gives $W_{i,t+1} = \alpha W_{i,t} - \eta w_{i,t} V_{i,t}$, since $\tau_{i,t}^D U_{i,t} = w_{i,t} V_{i,t}$. Applying the radial-component assumption along this trajectory gives

$$w_{i,t+1}^2 = \left(\alpha^2 + 2\eta\alpha\gamma + \eta^2\right) w_{i,t}^2, \qquad \frac{w_{i,t}}{w_{i,0}} = \left(\alpha^2 + 2\eta\alpha\gamma + \eta^2\right)^{t/2}. \tag{E.4}$$

Exponential growth occurs exactly when the constant multiplier of $w_{i,t}^2$ exceeds one. Since $\eta > 0$ and $1 - \eta\lambda > 0$,

$$\begin{aligned} \alpha^2 + 2\eta\alpha\gamma + \eta^2 > 1 \quad &\iff \quad 2\eta(1-\eta\lambda)\gamma > 2\eta\lambda - \eta^2(1+\lambda^2) \\ &\iff \quad \gamma > \frac{2\lambda - \eta(1+\lambda^2)}{2(1-\eta\lambda)}. \end{aligned} \tag{E.5}$$

For fixed λ, the threshold equals $\lambda + \eta(\lambda^2 - 1)/(2(1-\eta\lambda)) = \lambda + O(\eta)$ as $\eta \to 0$. This gives the small-learning-rate approximation $\gamma > \lambda$. □

E.4 Proof of Proposition 4

Fix a matrix type $\mathcal{G}$. Recall that w_i is the weight norm, r_i is the original u-w ratio, a_i is the first-order descent coefficient along $-V_i$, and $\widetilde{L}_i^{\mathcal{G}}$ is the adjusted directional curvature. By Hessian symmetry, $C_{ij} = C_{ji}$. Pairing the terms indexed by (i, j) and (j, i) in the definition of $\widetilde{L}_i^{\mathcal{G}}$ from Section 3.1 gives

$$\begin{aligned} \sum_{i\in\mathcal{G}} w_i^2 \widetilde{L}_i^{\mathcal{G}} &= \sum_{i,j\in\mathcal{G}} w_i w_j C_{ij}, \\ \sum_{i\in\mathcal{G}} (r_i w_i)^2 \widetilde{L}_i^{\mathcal{G}} &= \sum_{i,j\in\mathcal{G}} r_i r_j w_i w_j C_{ij}. \end{aligned} \tag{E.6}$$

Indeed, each distinct pair contributes $2w_i w_j C_{ij}$ to the first sum and $2r_i r_j w_i w_j C_{ij}$ to the second, using $r_i^2 r_j + r_j^2 r_i = r_i r_j (r_i + r_j)$ for the latter. Within-type alignment assigns the same positive ratio $\bar{r}_{\mathcal{G}}$ to every block in $\mathcal{G}$. This common factor cancels from $\mathcal{S}_{\mathcal{G}} = A_{\mathcal{G}}^2/(2B_{\mathcal{G}})$. Substituting Equation E.6 therefore yields

$$\mathcal{S}_{\mathcal{G}}(x^{\text{typed}}) = \frac{\left(\sum_{i\in\mathcal{G}} w_i a_i\right)^2}{2\sum_{i\in\mathcal{G}} w_i^2 \widetilde{L}_i^{\mathcal{G}}}, \qquad \mathcal{S}_{\mathcal{G}}(x^{\text{base}}) = \frac{\left(\sum_{i\in\mathcal{G}} r_i w_i a_i\right)^2}{2\sum_{i\in\mathcal{G}} (r_i w_i)^2 \widetilde{L}_i^{\mathcal{G}}}.$$

Use the weights $\pi_i = (r_i w_i)^2 / \sum_{j\in\mathcal{G}} (r_j w_j)^2$ and the weighted normalization from Definition 1. The assumptions on the original first-order and curvature terms imply

$$\left\langle \frac{a_i}{r_i w_i} \right\rangle_\pi = \frac{A_{\mathcal{G}}(x^{\text{base}})}{\sum_{i\in\mathcal{G}} (r_i w_i)^2} \neq 0, \qquad \left\langle \widetilde{L}^{\mathcal{G}} \right\rangle_\pi = \frac{B_{\mathcal{G}}(x^{\text{base}})}{\sum_{i\in\mathcal{G}} (r_i w_i)^2} > 0.$$

The preceding identities ensure that the weighted normalizations are well-defined. Rewriting the two scores using these normalizations gives

$$\frac{\mathcal{S}_{\mathcal{G}}(x^{\mathrm{typed}})}{\mathcal{S}_{\mathcal{G}}(x^{\mathrm{base}})} = \frac{\left\langle [r_i^{-1}]_\pi [a_i/(r_i w_i)]_\pi \right\rangle_\pi^2}{\left\langle ([r_i^{-1}]_\pi)^2 [\widetilde{L}_i^{\mathcal{G}}]_\pi \right\rangle_\pi}. \tag{E.7}$$

The denominator is positive, since

$$\left\langle ([r_i^{-1}]_\pi)^2 [\widetilde{L}_i^{\mathcal{G}}]_\pi \right\rangle_\pi = \frac{B_{\mathcal{G}}(x^{\mathrm{typed}})}{\bar{r}_{\mathcal{G}}^2 \langle r^{-1}\rangle_\pi^2 B_{\mathcal{G}}(x^{\mathrm{base}})} > 0.$$

It therefore suffices to show that the numerator exceeds the denominator. Expanding the denominator using the definitions of covariance and variance gives

$$\begin{aligned}
&\left\langle ([r_i^{-1}]_\pi)^2 [\widetilde{L}_i^{\mathcal{G}}]_\pi \right\rangle_\pi \\
&\quad = 1 + \mathrm{var}_\pi([r_i^{-1}]_\pi) + \mathrm{cov}_\pi\left(([r_i^{-1}]_\pi)^2, [\widetilde{L}_i^{\mathcal{G}}]_\pi\right) \\
&\quad < 1 + 2\,\mathrm{cov}_\pi\left([r_i^{-1}]_\pi, [a_i/(r_i w_i)]_\pi\right) \\
&\quad \le \left(1 + \mathrm{cov}_\pi\left([r_i^{-1}]_\pi, [a_i/(r_i w_i)]_\pi\right)\right)^2 \\
&\quad = \left\langle [r_i^{-1}]_\pi [a_i/(r_i w_i)]_\pi \right\rangle_\pi^2 .
\end{aligned} \tag{E.8}$$

The strict inequality is Equation C.8. The non-strict inequality follows by adding the nonnegative squared covariance. Thus the score ratio exceeds one, proving $\mathcal{S}_{\mathcal{G}}(x^{\mathrm{typed}}) > \mathcal{S}_{\mathcal{G}}(x^{\mathrm{base}})$.

For full alignment, we compare the two scores over all blocks in $\mathcal{I}$ under a shared learning rate. Pairing the Hessian terms over $\mathcal{I}$ gives the identities in Equation E.6 with sums over $\mathcal{I}$ and adjusted curvatures $\widetilde{L}_i^{\mathcal{I}}$. The common aligned ratio cancels from $A(x^{\mathrm{full}})^2/(2B(x^{\mathrm{full}}))$. Using the weights $\nu_i = (r_i w_i)^2 / \sum_{j\in\mathcal{I}} (r_j w_j)^2$ therefore gives

$$\frac{\mathcal{S}_{\mathrm{SL}}(x^{\mathrm{full}})}{\mathcal{S}_{\mathrm{SL}}(x^{\mathrm{base}})} = \frac{\left\langle [r_i^{-1}]_\nu [a_i/(r_i w_i)]_\nu \right\rangle_\nu^2}{\left\langle ([r_i^{-1}]_\nu)^2 [\widetilde{L}_i^{\mathcal{I}}]_\nu \right\rangle_\nu}.$$

Here the weighted means used to normalize $a_i/(r_i w_i)$ and $\widetilde{L}_i^{\mathcal{I}}$ are, respectively, $A(x^{\mathrm{base}})/\sum_{i\in\mathcal{I}}(r_i w_i)^2 \neq 0$ and $B(x^{\mathrm{base}})/\sum_{i\in\mathcal{I}}(r_i w_i)^2 > 0$. Moreover, $B(x^{\mathrm{full}}) > 0$ implies $\sum_{i\in\mathcal{I}} w_i^2 \widetilde{L}_i^{\mathcal{I}} > 0$, so the denominator of the score ratio is positive. Repeating the preceding covariance expansion with ν and $\widetilde{L}_i^{\mathcal{I}}$, Equation C.9 makes the denominator strictly smaller than the numerator. Consequently, $\mathcal{S}_{\mathrm{SL}}(x^{\mathrm{full}}) > \mathcal{S}_{\mathrm{SL}}(x^{\mathrm{base}})$. □

E.5 Proof of Proposition 5

We use the type-specific quadratic score derived in Appendix C.1 and compare the first-order vectors and curvature matrices of the two policies. Recall from Equation C.10 that $\pi_i^{\mathcal{G}}$ is proportional to the original update norm $r_i w_i$, and that $\bar{r}_{\mathcal{G}}$ is the common ratio assigned within type $\mathcal{G}$. To factor the overall update magnitude of each type out of its learning rate, define the positive quantities

$$q_{\mathcal{G}} := \sum_{i\in\mathcal{G}} r_i w_i, \qquad s_{\mathcal{G}} := \bar{r}_{\mathcal{G}} \sum_{i\in\mathcal{G}} w_i.$$

Here $q_{\mathcal{G}}$ and $s_{\mathcal{G}}$ are the sums of the original and aligned update norms in $\mathcal{G}$, respectively. Since $\langle r^{-1}\rangle_{\pi^{\mathcal{G}}} = (\sum_{i\in\mathcal{G}} w_i)/q_{\mathcal{G}}$, the coefficients of block i under the two policies satisfy

$$r_i w_i = q_{\mathcal{G}} \pi_i^{\mathcal{G}}, \qquad \bar{r}_{\mathcal{G}} w_i = s_{\mathcal{G}} \pi_i^{\mathcal{G}} [r_i^{-1}]_{\pi^{\mathcal{G}}}.$$

First-order and curvature terms. Recall the weighted means and covariances $g_{\mathcal{G}}$, $K_{\mathcal{G},\mathcal{H}}$, and $c_{\mathcal{G}}$ from Equation C.11. The covariance definition gives $\langle [r_i^{-1}]_{\pi^{\mathcal{G}}} a_i \rangle_{\pi^{\mathcal{G}}} = (1 + c_{\mathcal{G}}) g_{\mathcal{G}}$. For the curvature

terms, the product of the normalized inverse ratios has mean one under $\pi^{\mathcal{G}} \times \pi^{\mathcal{H}}$. Consequently, Equation C.12 gives

$$\begin{aligned}&\left\langle [r_i^{-1}]_{\pi^{\mathcal{G}}} [r_j^{-1}]_{\pi^{\mathcal{H}}} C_{ij} \right\rangle_{\pi^{\mathcal{G}} \times \pi^{\mathcal{H}}} \\ &\quad = K_{\mathcal{G},\mathcal{H}} + \mathrm{cov}_{\pi^{\mathcal{G}} \times \pi^{\mathcal{H}}} \left([r_i^{-1}]_{\pi^{\mathcal{G}}} [r_j^{-1}]_{\pi^{\mathcal{H}}}, C_{ij} \right) \\ &\quad = (\mathbf{DKD} - \mathbf{\Gamma})_{\mathcal{G},\mathcal{H}}.\end{aligned}$$

Substituting the block coefficients into the first-order and curvature terms from Section 3.3 therefore yields

$$\begin{aligned} A_{\mathcal{G}}(x^{\mathrm{base}}) &= q_{\mathcal{G}} g_{\mathcal{G}}, & B_{\mathcal{G},\mathcal{H}}(x^{\mathrm{base}}) &= q_{\mathcal{G}} q_{\mathcal{H}} K_{\mathcal{G},\mathcal{H}}, \\ A_{\mathcal{G}}(x^{\mathrm{typed}}) &= s_{\mathcal{G}}(1 + c_{\mathcal{G}}) g_{\mathcal{G}}, & B_{\mathcal{G},\mathcal{H}}(x^{\mathrm{typed}}) &= s_{\mathcal{G}} s_{\mathcal{H}} (\mathbf{DKD} - \mathbf{\Gamma})_{\mathcal{G},\mathcal{H}}. \end{aligned} \tag{E.9}$$

Optimized scores. For the original optimizer, express the quadratic model using the rescaled learning rates $\zeta_{\mathcal{G}} = q_{\mathcal{G}} \eta_{\mathcal{G}}$. Equation E.9 gives first-order vector $\mathbf{g}$ and curvature matrix $\mathbf{K}$ in these coordinates. For within-type alignment, using $\zeta_{\mathcal{G}} = s_{\mathcal{G}} \eta_{\mathcal{G}}$ instead gives first-order vector $\mathbf{Dg}$ and curvature matrix $\mathbf{DKD} - \mathbf{\Gamma}$. These changes of variables are invertible because $q_{\mathcal{G}}, s_{\mathcal{G}} > 0$. They therefore preserve the maximum over learning-rate vectors in $\mathbb{R}^N$ used to define $\mathcal{S}_{\mathrm{TL}}$.

Since $c_{\mathcal{G}} \neq -1$ for every type, $\mathbf{D}$ is invertible. Write $\mathbf{R} := \mathbf{D}^{-1} \mathbf{\Gamma} \mathbf{D}^{-1}$, so that $\mathbf{DKD} - \mathbf{\Gamma} = \mathbf{D}(\mathbf{K} - \mathbf{R})\mathbf{D}$. By assumption, the aligned policy's curvature matrix $\mathbf{DKD} - \mathbf{\Gamma}$ is positive definite. Since $\mathbf{D}$ is invertible,

$$\mathbf{K} - \mathbf{R} = \mathbf{D}^{-1}(\mathbf{DKD} - \mathbf{\Gamma})\mathbf{D}^{-1} \succ 0.$$

Substituting the rescaled vectors and matrices into the score from Appendix C.1 gives

$$\begin{aligned} \mathcal{S}_{\mathrm{TL}}(x^{\mathrm{base}}) &= \frac{1}{2} \mathbf{g}^\top \mathbf{K}^{-1} \mathbf{g}, \\ \mathcal{S}_{\mathrm{TL}}(x^{\mathrm{typed}}) &= \frac{1}{2} (\mathbf{Dg})^\top (\mathbf{DKD} - \mathbf{\Gamma})^{-1} (\mathbf{Dg}) = \frac{1}{2} \mathbf{g}^\top (\mathbf{K} - \mathbf{R})^{-1} \mathbf{g}. \end{aligned} \tag{E.10}$$

Generalized eigenmodes. The generalized eigenpairs in the proposition satisfy $\mathbf{R}\mathbf{v}_k = \rho_k \mathbf{K}\mathbf{v}_k$ and $\mathbf{v}_k^\top \mathbf{K} \mathbf{v}_\ell = \mathbf{1}\{k = \ell\}$. Symmetry of $\mathbf{R}$ and positive definiteness of $\mathbf{K}$ ensure that these eigenvectors form a complete real basis. Moreover,

$$\mathbf{v}_k^\top (\mathbf{K} - \mathbf{R}) \mathbf{v}_\ell = (1 - \rho_\ell) \mathbf{1}\{k = \ell\}.$$

Positive definiteness of $\mathbf{K} - \mathbf{R}$ therefore gives $1 - \rho_k > 0$ for every k. The two quadratic forms are diagonal in this basis, so

$$\mathbf{K}^{-1} = \sum_{k=1}^{N} \mathbf{v}_k \mathbf{v}_k^\top, \qquad (\mathbf{K} - \mathbf{R})^{-1} = \sum_{k=1}^{N} \frac{\mathbf{v}_k \mathbf{v}_k^\top}{1 - \rho_k}.$$

Substituting into Equation E.10 and subtracting gives

$$\mathcal{S}_{\mathrm{TL}}(x^{\mathrm{typed}}) - \mathcal{S}_{\mathrm{TL}}(x^{\mathrm{base}}) = \frac{1}{2} \sum_{k=1}^{N} \frac{\rho_k}{1 - \rho_k} (\mathbf{v}_k^\top \mathbf{g})^2. \tag{E.11}$$

The positive factor $1/2$ does not affect the sign, proving the equivalence in Equation C.13.

To connect this eigenmode calculation to the matched-first-order interpretation used in Section 3.3, fix an eigenmode k and choose the learning rates

$$\eta_{\mathcal{G}}^{\mathrm{base}} = \frac{v_{k,\mathcal{G}}}{q_{\mathcal{G}}}, \qquad \eta_{\mathcal{G}}^{\mathrm{typed}} = \frac{v_{k,\mathcal{G}}}{s_{\mathcal{G}}(1 + c_{\mathcal{G}})}, \qquad \mathcal{G} \in \mathfrak{G}. \tag{E.12}$$

By Equation E.9, both choices have first-order term $\mathbf{v}_k^\top \mathbf{g}$. Their quadratic curvature terms are

$$\mathbf{v}_k^\top \mathbf{K} \mathbf{v}_k = 1 \qquad \text{and} \qquad \mathbf{v}_k^\top (\mathbf{K} - \mathbf{R}) \mathbf{v}_k = 1 - \rho_k,$$

respectively. Thus, under the same first-order decrease, ρ_k is the fractional curvature reduction from within-type alignment. □

E.6 Proof of Corollary 1

We apply Proposition 4 to SGD without momentum. Since $U_i = G_i$, the unit update direction is $V_i = G_i/\|G_i\|_F$, and the first-order descent coefficient is $a_i = \langle G_i, V_i\rangle_F = \|G_i\|_F$. Consequently,

$$r_i w_i = \|G_i\|_F, \qquad r_i^{-1} = \frac{w_i}{\|G_i\|_F}, \qquad \frac{a_i}{r_i w_i} = 1.$$

For within-type alignment, the weights in Proposition 4 become $\pi_i = \|G_i\|_F^2 / \sum_{j\in\mathcal{G}} \|G_j\|_F^2$. The required condition $A_{\mathcal{G}}(x^{\mathrm{base}}) \neq 0$ holds because $A_{\mathcal{G}}(x^{\mathrm{base}}) = \sum_{i\in\mathcal{G}} \|G_i\|_F^2 > 0$. The two curvature terms $B_{\mathcal{G}}(x^{\mathrm{typed}})$ and $B_{\mathcal{G}}(x^{\mathrm{base}})$ are positive by the assumptions of the corollary. Moreover, $[a_i/(r_i w_i)]_\pi = 1$, so the covariance in the first-order descent gain vanishes:

$$\mathrm{cov}_\pi\left([r_i^{-1}]_\pi, [a_i/(r_i w_i)]_\pi\right) = \mathrm{cov}_\pi\left([w_i/\|G_i\|_F]_\pi, 1\right) = 0. \tag{E.13}$$

Substituting this and $r_i^{-1} = w_i/\|G_i\|_F$ into Equation C.8 reduces the sufficient condition to

$$-\mathrm{cov}_\pi\left(\left([w_i/\|G_i\|_F]_\pi\right)^2, [\widetilde{L}_i^{\mathcal{G}}]_\pi\right) > \mathrm{var}_\pi([w_i/\|G_i\|_F]_\pi),$$

which is Equation D.1. Proposition 4 therefore yields $\mathcal{S}_{\mathcal{G}}(x^{\mathrm{typed}}) > \mathcal{S}_{\mathcal{G}}(x^{\mathrm{base}})$.

For full alignment, the comparison includes all SGD-updated blocks in $\mathcal{I}$. The weights become $\nu_i = \|G_i\|_F^2 / \sum_{j\in\mathcal{I}} \|G_j\|_F^2$, and the adjusted curvatures are $\widetilde{L}_i^{\mathcal{I}}$, which include interactions across all these blocks. Again, $A(x^{\mathrm{base}}) = \sum_{i\in\mathcal{I}} \|G_i\|_F^2 > 0$, and $[a_i/(r_i w_i)]_\nu = 1$ makes the first-order covariance zero. Thus Equation C.9 reduces to

$$-\mathrm{cov}_\nu\left(\left([w_i/\|G_i\|_F]_\nu\right)^2, [\widetilde{L}_i^{\mathcal{I}}]_\nu\right) > \mathrm{var}_\nu([w_i/\|G_i\|_F]_\nu),$$

which is Equation D.2. Together with the assumed positivity of $B(x^{\mathrm{full}})$ and $B(x^{\mathrm{base}})$, this condition allows us to apply Proposition 4 and conclude that $\mathcal{S}_{\mathrm{SL}}(x^{\mathrm{full}}) > \mathcal{S}_{\mathrm{SL}}(x^{\mathrm{base}})$. □