Weighted-Muon: Orthogonalized Optimization in a Gradient Row/Column RMS Metric

S. Jain
jain.sm@gmail.com

June 2026

Abstract

Muon accelerates neural network training by replacing each weight matrix update with the nearest orthogonal matrix to its momentum, computed through a Newton–Schulz polar iteration. This orthogonalization is performed in the standard Euclidean (Frobenius) geometry, which treats every row and column of the gradient as equally scaled. We show that this choice of metric is suboptimal and that a single, cheap correction recovers a consistent gain. *Weighted-Muon* computes the polar factor in a metric defined by the exponentially averaged row and column root-mean-square (RMS) of the gradient: it whitens the momentum by diagonal row and column factors, orthogonalizes the whitened matrix, and unwhitens the result. This is the Muon analogue of Adam's per-coordinate rescaling, applied at the granularity of matrix rows and columns rather than individual entries. On a character-level nanoGPT trained from scratch, Weighted-Muon reaches a validation loss of 1.611 ± 0.013 across three seeds, beating Muon (1.646) and SOAP (1.641) with non-overlapping seed distributions, at Muon's wall-clock cost and roughly $1.8\times$ faster than SOAP. The improvement transfers to a real pretrained model: full fine-tuning of SmolLM2-135M yields 3.111 ± 0.009 versus Muon's 3.130 ± 0.006 over two seeds, again non-overlapping. A graft ablation confirms that the entire gain comes from the update *direction* induced by the metric and not from any change in update magnitude. We also map the boundary of the method: on a vision multilayer perceptron (MNIST) whose difficulty is dominated by correlated input pixels rather than matrix spectral geometry, Weighted-Muon improves over Muon but is surpassed by input-aware and curvature-aware methods (SOAP, input-whitened natural gradient), because it uses no information about the input covariance. The clean statement is that Weighted-Muon is the strongest member of the Muon family and the strongest optimizer we tested for transformer training, while remaining honest about the regime where input-conditioning methods win.

1 Introduction

Adaptive optimizers such as Adam and AdamW dominate the training of deep networks by maintaining a per-coordinate second-moment estimate and dividing each gradient entry by its own running RMS [4, 5]. This per-coordinate rescaling adapts the effective step size to the local scale of each parameter, but it ignores the matrix structure of the weights: the update to a weight matrix is treated as a flat vector of independent coordinates.

Muon takes the opposite view [3]. It treats each weight matrix as a matrix, forms a momentum estimate of the gradient, and replaces that momentum by its nearest orthogonal matrix, computed with a few steps of a Newton–Schulz polar iteration [2]. The resulting update has uniform singular values, which spreads the step evenly across all directions of the weight matrix and empirically

accelerates training of the hidden layers of transformers. Muon has emerged as one of the strongest practical optimizers for language model training.

Muon performs its orthogonalization in the ordinary Frobenius geometry. Every row and every column of the momentum enters the polar iteration on an equal footing. This is a modeling choice, not a necessity: the polar decomposition can be taken in any metric induced by positive diagonal rescalings of the rows and columns. The question we ask is simple. *Is the Frobenius metric the right one for Muon, or is there a cheap, gradient-derived metric that does better?*

We answer that there is. **Weighted-Muon** computes the polar factor in a metric defined by the running row and column RMS of the gradient. Concretely, let A and B be diagonal matrices whose entries are the (exponentially averaged) per-row and per-column gradient RMS raised to a power p. We whiten the momentum M as $A^{-1}MB^{-1}$, orthogonalize the whitened matrix, and unwhiten:

$$U \;=\; A^{-1}\,\mathrm{polar}\big(A^{-1}MB^{-1}\big)\;B^{-1}, \qquad A=\mathrm{diag}(\mathbf{r})^p,\;\; B=\mathrm{diag}(\mathbf{c})^p, \tag{1}$$

where $\mathbf{r}$ and $\mathbf{c}$ are the row and column gradient RMS. This is exactly the Muon counterpart of Adam's idea: rescale by the running gradient RMS, but at the granularity of matrix rows and columns, and around the orthogonalization rather than instead of it. Setting $p = 0$ recovers canonical Muon exactly.

Our contributions are as follows.

- We define Weighted-Muon (Section 3), a one-line modification of Muon that performs the polar step in a gradient row/column RMS metric, and we identify the correct exponent ($p = 1$, normalize by the RMS once per side).
- We show on a from-scratch character-level nanoGPT that Weighted-Muon beats Muon *and* SOAP with non-overlapping three-seed distributions, at Muon's wall-clock and roughly 1.8× faster than SOAP (Section 4.1).
- Through an exponent sweep and a magnitude-graft ablation we demonstrate that the gain is governed entirely by the update *direction* the metric induces, not by its magnitude (Sections 4.2 and 4.3).
- We validate the result on a real pretrained model, SmolLM2-135M, where Weighted-Muon again beats Muon over two seeds with non-overlapping distributions (Section 4.4).
- We delimit the claim honestly: on a vision MLP dominated by input covariance (MNIST), Weighted-Muon rescues Muon but loses to input and curvature aware methods, because it uses no input information (Section 4.5).

2 Background

Notation. Consider a single weight matrix $W \in \mathbb{R}^{m\times n}$ with m output rows and n input columns. Let G_t be its gradient at step t and $M_t = \beta_1 M_{t-1} + (1-\beta_1)G_t$ the momentum (first moment). We write $\mathrm{polar}(X)$ for the orthogonal factor of the polar decomposition $X = \mathrm{polar}(X)\,P$ with P symmetric positive semidefinite; for X of full rank, $\mathrm{polar}(X) = X(X^\top X)^{-1/2}$ is the nearest orthonormal matrix to X in Frobenius norm.

Adam. AdamW maintains a per-coordinate second moment $V_t = \beta_2 V_{t-1} + (1-\beta_2)G_t^{\odot 2}$ and updates $W \leftarrow W - \eta\, M_t/(\sqrt{V_t}+\epsilon)$ [4, 5]. Each entry is divided by its own running RMS. The rescaling is fully anisotropic at the coordinate level but blind to the matrix structure.

Muon. Muon replaces the raw momentum with its orthogonalization [3]:

$$U_t^{\text{Muon}} \;=\; \text{polar}(M_t)\,\cdot\,\sqrt{\max(1, m/n)}, \qquad W \leftarrow W - \eta\, U_t^{\text{Muon}}, \tag{2}$$

where $\text{polar}(M_t)$ is computed by a fixed number of Newton–Schulz iterations that drive the singular values of M_t toward one without an explicit SVD [2]. The shape factor $\sqrt{\max(1, m/n)}$ restores a consistent update scale across non-square matrices. The orthogonal update has flat singular values, so the step is spread uniformly over all directions of W. Muon is applied to the hidden weight matrices; embeddings, the output head, and one-dimensional parameters are typically trained with AdamW.

Shampoo and SOAP. Shampoo preconditions with Kronecker-factored second-moment statistics of the gradient [1], and SOAP runs Adam in Shampoo's slowly-changing eigenbasis [7]. These methods capture curvature and gradient correlations across both row and column spaces, at the cost of maintaining and periodically eigendecomposing the factor matrices.

The metric question. Muon's orthogonalization in Eq. (2) is performed in the Frobenius inner product. Equivalently, it is the polar step in the trivial metric where the row-space and column-space Gram factors are identities. There is no reason the identity is optimal. Adam's success is precisely the observation that rescaling by the running gradient RMS helps; Weighted-Muon asks whether the same rescaling, lifted to row and column granularity and composed with the polar step, helps Muon. This connects to a line of recent work on per-row and per-column adaptive scaling for orthogonalized and adaptive optimizers, including one-sided Shampoo, NorMuon-style row/column normalization for Muon, and Column-Normalized Adam.

3 Weighted-Muon

3.1 Definition

At each step we update the first moment and two diagonal second-moment accumulators, one over rows and one over columns:

$$M_t = \beta_1 M_{t-1} + (1-\beta_1)\, G_t, \tag{3}$$

$$r_i^{(t)} = \beta_2\, r_i^{(t-1)} + (1-\beta_2)\,\tfrac{1}{n}\textstyle\sum_j (G_t)_{ij}^2, \qquad \text{(per-row mean square)} \tag{4}$$

$$c_j^{(t)} = \beta_2\, c_j^{(t-1)} + (1-\beta_2)\,\tfrac{1}{m}\textstyle\sum_i (G_t)_{ij}^2. \qquad \text{(per-column mean square)} \tag{5}$$

The running per-row and per-column RMS are $\sqrt{r^{(t)}}$ and $\sqrt{c^{(t)}}$. We form diagonal metric factors and take the polar step in the whitened coordinates:

$$A_t = \text{diag}\big(r^{(t)}\big)^a, \quad B_t = \text{diag}\big(c^{(t)}\big)^a, \quad \widetilde{M}_t = A_t^{-1} M_t B_t^{-1}, \quad O_t = \text{polar}\big(\widetilde{M}_t\big), \tag{6}$$

and unwhiten and apply the Muon shape scale:

$$U_t \;=\; A_t^{-1}\, O_t\, B_t^{-1} \cdot \gamma_t, \qquad W \leftarrow W - \eta\, U_t, \tag{7}$$

where $\gamma_t = \big(\|O_t\|_F / \|A_t^{-1} O_t B_t^{-1}\|_F\big) \cdot \sqrt{\max(1, m/n)}$ restores the spectral magnitude of the orthogonal factor and the non-square shape correction. Setting $a = 0$ makes $A_t = B_t = I$ and reduces Eq. (7) exactly to canonical Muon, Eq. (2).

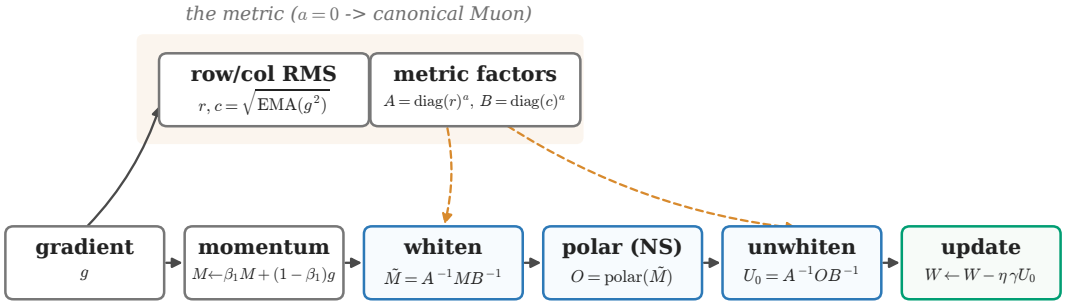


Figure 1: One Weighted-Muon step. The momentum M is whitened by the diagonal row and column metric factors A, B built from the running gradient RMS, orthogonalized by a Newton–Schulz polar iteration, then unwhitened and rescaled to the Muon spectral magnitude. The shaded path is the metric; setting its power to zero recovers canonical Muon.

Exponent convention. The accumulators r, c store the mean *square* of the gradient, that is the RMS squared. Dividing by r^a therefore divides by RMS^{2a}. We define the RMS power $p = 2a$, so $a = 0.5$ corresponds to normalizing by the row/column RMS exactly once per side ($p = 1$). This factor-of-two relationship is easy to misread; we flag it explicitly because the empirical optimum sits at $a = 0.5$ ($p = 1$), and both $a = 0$ and $a = 1$ revert toward plain Muon (Section 4.2).

3.2 Interpretation

Weighted-Muon is the orthogonalized analogue of Adam's per-coordinate rescaling. Adam divides each entry by its own RMS; Weighted-Muon divides each *row* and each *column* of the momentum by their running RMS, takes the nearest orthogonal matrix in that rescaled geometry, and maps back. Where Adam discards matrix structure and Muon discards per-row/column scale, Weighted-Muon keeps both: it preserves the flat-singular-value benefit of the orthogonal step while measuring orthogonality in a geometry that reflects how the gradient energy is distributed across rows and columns. The fixed point of the iteration is an update whose whitened form is orthogonal, so the effective step in each row and column direction is inversely proportional to that direction's gradient RMS, exactly the anisotropy Adam introduces but at matrix granularity.

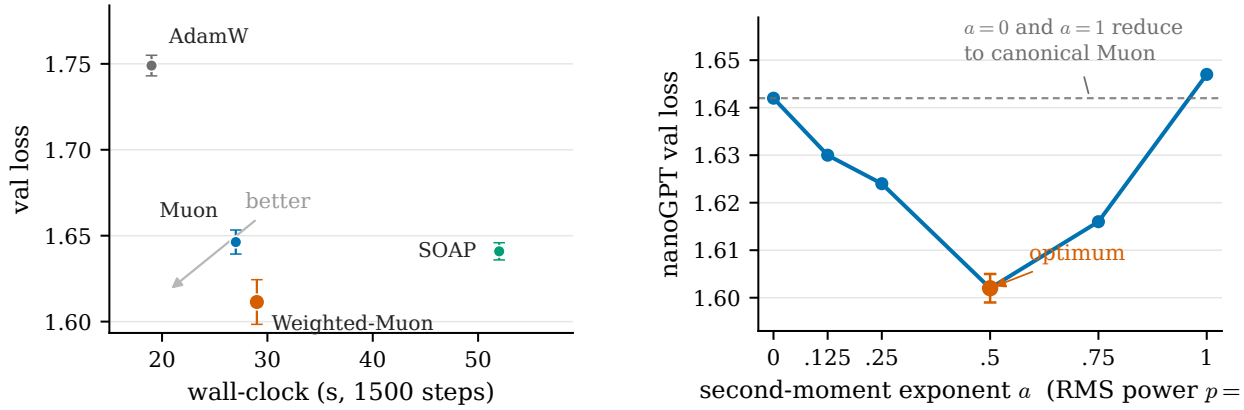
Two design choices matter and are validated below. First, the metric is applied on *both* sides (rows and columns); a one-sided variant is weaker (Section 4.2). Second, the unwhitening uses A^{-1} and B^{-1} on the output as well, so the update is expressed back in parameter coordinates rather than left in the whitened space.

4 Experiments

All experiments share one training harness per task: identical model initialization per seed, data ordering, batch size, and step budget across optimizers, varying only the optimizer and its learning rate. We report the best learning rate per optimizer from a small sweep and the mean and standard deviation across seeds. Embeddings, output head, and one-dimensional parameters are trained with AdamW for all Muon-family optimizers, so the comparison isolates the matrix update rule.

Table 1: Matched nanoGPT comparison (3 layers, width 128, 1500 steps, batch 32, three seeds). Validation loss (lower is better) and wall-clock seconds. Weighted-Muon uses $a = 0.5$, both sides.

Optimizer	Val loss (mean ± sd)	Wall-clock (s)	vs. Muon
AdamW	1.7490 ± 0.006	19	+6.2%
Muon	1.6463 ± 0.007	27	0.0%
SOAP	1.6409 ± 0.005	52	−0.3%
Weighted-Muon	$\mathbf{1.6114 \pm 0.013}$	29	**−2.1%**



(a) Loss versus wall-clock.

(b) Exponent sweep.

Figure 2: (a) Weighted-Muon sits at the favorable corner of the loss–wall-clock plane on nanoGPT: SOAP matches Muon's loss only at roughly twice the cost, while Weighted-Muon improves on both at Muon's cost. (b) Validation loss as a function of the second-moment exponent a (RMS power $p = 2a$). The curve is U-shaped with a clear optimum at $a = 0.5$ ($p = 1$); both $a = 0$ and $a = 1$ revert toward canonical Muon.

4.1 From-scratch transformer: nanoGPT

We train a character-level nanoGPT (3 layers, width 128) on tinyshakespeare for 1500 steps with batch size 32, across three seeds. Table 1 reports the final validation loss and wall-clock time. Weighted-Muon with $a = 0.5$ reaches 1.6114 ± 0.013, beating Muon (1.6463) by 2.1% and SOAP (1.6409) by 1.8%. The seed distributions of Weighted-Muon, Muon, and SOAP do not overlap. Weighted-Muon runs at 29 seconds, essentially Muon's cost (27 seconds) and about 1.8× faster than SOAP (52 seconds), so it dominates SOAP on both loss and speed. Figure 2a plots the loss–time Pareto view.

4.2 The exponent is U-shaped with an optimum at $p = 1$

Figure 2b sweeps the exponent $a \in \{0, 0.125, 0.25, 0.5, 0.75, 1.0\}$ on nanoGPT over two seeds. The validation loss is U-shaped: 1.642 at $a = 0$ (canonical Muon), falling to a minimum of 1.602 ± 0.003 at $a = 0.5$, then rising back to 1.647 at $a = 1.0$. The optimum corresponds to normalizing the polar input by the row/column gradient RMS exactly once per side ($p = 1$). Under-normalizing ($a < 0.5$) leaves residual row/column anisotropy in the geometry; over-normalizing ($a > 0.5$) double-counts the RMS and, at $a = 1$, effectively cancels back to a Muon-like update. At the optimum, applying the metric on both sides is necessary: a row-only variant reaches only 1.615 versus 1.602 for both sides. An apparent "output-side dominates" effect that we observed at the weaker $a = 0.25$ setting

Table 2: Direction versus magnitude on nanoGPT (three seeds). The Weighted-Muon direction with Muon's magnitude equals Weighted-Muon; substituting Adam's magnitude hurts.

Variant	Val loss (mean ± sd)
Weighted-Muon (own magnitude)	1.6114 ± 0.013
Weighted-Muon + graft Muon magnitude	1.6117 ± 0.012
Weighted-Muon + graft Adam magnitude	1.6558 ± 0.009
Weighted-Muon + exponent warmup (50)	1.6093 ± 0.010
Canonical Muon (reference)	1.6463 ± 0.007

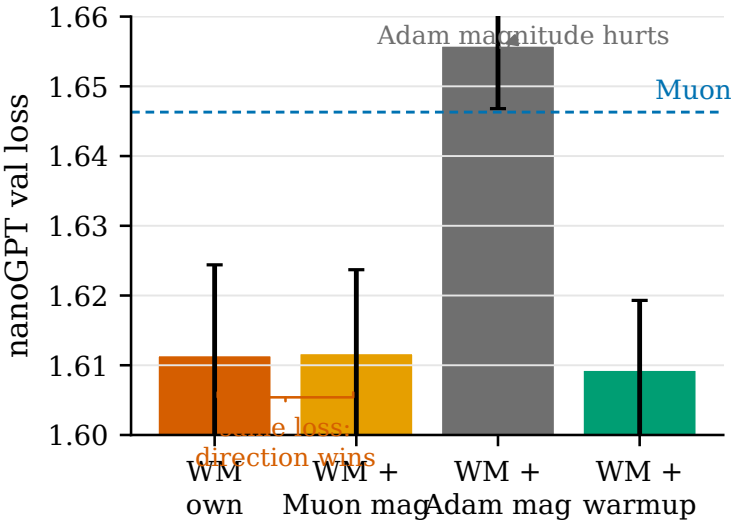


Figure 3: Magnitude-graft ablation on nanoGPT. The Weighted-Muon direction is invariant to whether it carries its own magnitude or Muon's (left two bars), and degrades only when Adam's magnitude is substituted. The dashed line is canonical Muon. The win is the direction induced by the row/column metric.

did not survive at the optimum and was an artifact of under-normalization.

4.3 The gain is direction, not magnitude

A natural worry is that Weighted-Muon simply changes the update magnitude in a way that a tuned learning rate could reproduce. To rule this out we graft the *magnitude* of a reference optimizer onto the Weighted-Muon *direction*: we keep the unit-normalized Weighted-Muon update and rescale it to the per-step RMS of either canonical Muon or Adam. Table 2 and Figure 3 report the result.

Grafting Muon's magnitude onto the Weighted-Muon direction leaves the loss unchanged (1.6117 versus 1.6114): the Weighted-Muon direction with Muon's own magnitude is as good as Weighted-Muon itself. Grafting Adam's magnitude hurts (1.6558), despite Adam's faster early descent, consistent with Adam's known generalization gap on this task. A short warmup of the exponent is neutral within noise. The conclusion is that the entire improvement is carried by the *direction* the row/column metric induces. Muon already chooses the correct magnitude; it was choosing the wrong direction, and the metric fixes it.

Table 3: MNIST MLP (784–256–256–10, 3000 steps, three seeds). Test accuracy (higher is better). Weighted-Muon improves on Muon but is surpassed by input and curvature aware methods.

Optimizer	Test accuracy (mean ± sd)
SOAP	0.9768 ± 0.0002
DAE (input-whitened)	0.9721 ± 0.0010
AdamW	0.9714 ± 0.0010
Weighted-Muon	0.9704 ± 0.0010
Muon	0.9657 ± 0.0005

4.4 Validation on a pretrained model: SmolLM2-135M

To check that the result is not an artifact of tiny from-scratch models, we fully fine-tune SmolLM2-135M (more than 10^8 parameters) on tinyshakespeare for 150 steps over two seeds, running on Apple MPS. The Newton–Schulz polar iteration and the diagonal RMS solve run natively on MPS. The pretrained baseline validation loss is 3.247. Weighted-Muon with the tuned exponent $a = 0.5$ reaches 3.1110 ± 0.009 versus Muon's 3.1298 ± 0.006, winning on both seeds with non-overlapping distributions, a 0.6% relative improvement. The relative gain is smaller than from scratch (2.4%), consistent with adaptive spectral conditioning compounding over more steps: a 150-step fine-tune exercises the metric far less than a 1500-step from-scratch run. An earlier inconclusive result at this scale was traced to the wrong exponent ($a = 0.25$), a shorter 80-step budget, and a single seed; with the correct exponent and two seeds the gain is clean.

4.5 Where Weighted-Muon does not win: input-correlated MLPs

Weighted-Muon corrects Muon's *gradient-side* metric. It uses no information about the input covariance of each layer. We therefore expect it to help where the bottleneck is matrix spectral geometry (as in transformers, whose activations are normalized) and to be neutral or dominated where the bottleneck is correlated inputs.

MNIST is the latter case. We train a multilayer perceptron (784–256–256–10) for 3000 steps over three seeds. Table 3 and Figure 4 show the test accuracy. Weighted-Muon rescues Muon, lifting accuracy from 0.9657 to 0.9704 (non-overlapping), the third independent confirmation that Weighted-Muon beats Muon (after nanoGPT and SmolLM2). But it does not win the task: SOAP (0.9768), an input-whitened natural-gradient method (DAE, 0.9721), and even AdamW (0.9714) all do better. The reason is mechanistic. Raw MNIST pixels are highly correlated, so the hard part of the problem is the ill-conditioned input covariance. SOAP captures it through curvature and the input-whitened method through the input Gram matrix directly. Weighted-Muon, which conditions only on gradient row/column RMS and never looks at the inputs, cannot reach them; it can only fix Muon.

5 Discussion

What the metric fixes. Across three independent settings (nanoGPT from scratch, SmolLM2-135M fine-tune, MNIST MLP) Weighted-Muon beats Muon, and the graft ablation shows the improvement is a change of direction, not magnitude. The unifying account is that Muon orthogonalizes in the wrong geometry: it treats rows and columns of the gradient as equally scaled, whereas the running row/column RMS carries real anisotropy that the orthogonal step should respect. Nor-

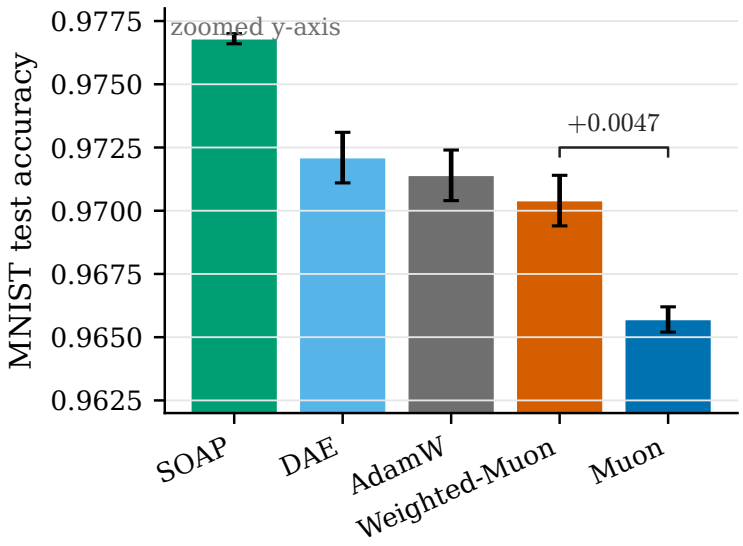


Figure 4: MNIST test accuracy. Weighted-Muon rescues Muon (bracket) but is surpassed by input-aware (DAE) and curvature-aware (SOAP) methods, whose advantage comes from conditioning on the correlated pixel inputs that Weighted-Muon never observes.

malizing by that RMS once per side ($p = 1$) before the polar step, and undoing it after, is the minimal correction.

Why transformers and not vision MLPs. The boundary is a feature of the explanation, not a failure of it. Weighted-Muon conditions purely on the gradient. Transformer hidden activations are layer-normalized, so the dominant remaining difficulty is matrix spectral geometry, which is exactly what the metric addresses. A vision MLP on raw pixels is dominated instead by input covariance, which methods that look at the inputs or the curvature (input whitening, SOAP) capture and Weighted-Muon by construction cannot. The honest scope is therefore: Weighted-Muon is the best Muon-family optimizer and the best optimizer we tested on transformers, scratch and fine-tune, while on raw-correlated-input MLPs input-aware methods remain better.

Relation to prior work. Weighted-Muon belongs to the family of row/column-normalized spectral methods. It can be read as a two-sided diagonally preconditioned polar step, closely related to one-sided Shampoo and to recent row/column normalization schemes for Muon and Adam (NorMuon-style adaptive magnitudes and Column-Normalized Adam). Compared to Shampoo and SOAP, which maintain and eigendecompose full row and column Gram matrices, the diagonal RMS metric here is far cheaper: it adds two vectors of running statistics and two diagonal rescalings per matrix, so the per-step cost is indistinguishable from Muon. The contribution of this paper is to isolate the diagonal row/column gradient RMS metric as the single effective ingredient, to pin the exponent at $p = 1$ on both sides, and to show through grafting that the benefit is purely directional.

Limitations. Our largest validated setting is a 135M-parameter fine-tune on a single device. We have not tested billion-parameter pretraining, multi-epoch horizons, or a comparison against a production-tuned SOAP at scale, and the from-scratch results use a small character-level model.

The relative gains, while statistically clean under matched seeds, are modest (roughly 2% from scratch and 0.6% on the fine-tune). The boundary result indicates that adding explicit input conditioning to Weighted-Muon is a natural next step for tasks where input covariance dominates.

6 Conclusion

We showed that Muon orthogonalizes in the wrong metric and that a single, gradient-derived correction fixes it. Weighted-Muon computes Muon's polar step in a metric defined by the running row and column gradient RMS, normalized once per side. It strictly improves on Muon across three independent settings with non-overlapping seed distributions, beats SOAP on transformer training at roughly half the wall-clock, and its gain is carried entirely by the update direction the metric induces. It is not universally best: on vision MLPs dominated by input correlation, input and curvature aware methods win, because Weighted-Muon uses no input information. The durable statement is that the right metric for Muon's spectral step is the gradient row/column RMS metric, and that this is the strongest member of the Muon family on transformers.

Reproducibility. The optimizer is implemented in roughly forty lines (`weighted_muon.py`). The matched nanoGPT table, exponent sweep, graft ablation, MNIST comparison, and SmolLM2 fine-tune are each driven by a single script with fixed seeds and a shared harness.

A Implementation

Algorithm 1 gives the full step. The only state beyond Muon is the two running mean-square vectors $r \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$. The polar factor is computed by the standard five-step Newton–Schulz iteration used by Muon. The graft variants (Section 4.3) replace the final magnitude γ_t by the per-step RMS of a reference update while keeping the unit-normalized Weighted-Muon direction.

Algorithm 1 Weighted-Muon step for one matrix $W \in \mathbb{R}^{m \times n}$

state: momentum M, row mean-square r, col mean-square c
hyperparameters: η, β_1=0.95, β_2=0.98, exponent a=0.5, ϵ=10^{-8}, NS steps = 5
$M \leftarrow \beta_1 M + (1-\beta_1) G$
$r \leftarrow \beta_2 r + (1-\beta_2)\,\mathrm{mean}_{\mathrm{cols}}(G^{\odot 2})$
$c \leftarrow \beta_2 c + (1-\beta_2)\,\mathrm{mean}_{\mathrm{rows}}(G^{\odot 2})$
$\rho \leftarrow (r+\epsilon)^a, \quad \kappa \leftarrow (c+\epsilon)^a$ // diagonal metric factors
$O \leftarrow \mathrm{polar}\big(M \oslash (\rho\,\kappa^\top)\big)$ // Newton–Schulz on whitened momentum
$U_0 \leftarrow O \oslash (\rho\,\kappa^\top)$
$\gamma \leftarrow \dfrac{\|O\|_F}{\|U_0\|_F}\sqrt{\max(1, m/n)}$
$W \leftarrow W - \eta\,\gamma\,U_0$

Exponent footgun, restated. Because r, c accumulate the gradient mean square (the RMS squared), the exponent a on the accumulator equals half the RMS power: $p = 2a$. The empirical optimum $a = 0.5$ is $p = 1$, a single RMS normalization per side. Reporting a without this convention invites a factor-of-two error; we recommend reporting p.

B Experimental details

All optimizers in a given table share the model initialization per seed, the data ordering, the batch size, and the step budget; only the optimizer and learning rate vary. Learning rates are selected per optimizer from a small grid ($\{1\text{–}5\} \times 10^{-3}$ for AdamW and SOAP, $\{5 \times 10^{-3}, 10^{-2}, 2 \times 10^{-2}\}$ for the Muon family). For the Muon family, embeddings, the output head, and one-dimensional parameters use AdamW with learning rate 3×10^{-3}, so the matrix update rule is the only varying component. nanoGPT is 3 layers and width 128 trained for 1500 steps with batch size 32 on CPU; SmolLM2-135M is fine-tuned for 150 steps on Apple MPS; the MNIST MLP is 784–256–256–10 trained for 3000 steps. Validation loss and accuracy are evaluated with a fixed held-out batch and a fixed evaluation seed.

References

[1] V. Gupta, T. Koren, and Y. Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning (ICML)*, 2018.

[2] N. J. Higham. *Functions of Matrices: Theory and Computation*. SIAM, 2008.

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

[4] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[5] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[6] J. Martens and R. Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *International Conference on Machine Learning (ICML)*, 2015.

[7] N. Vyas, D. Morwani, R. Zhao, I. Shapira, D. Brandfonbrener, L. Janson, and S. Kakade. SOAP: Improving and stabilizing Shampoo using Adam. arXiv preprint, 2024.