FIRMO: ROBUST MATRIX-AWARE OPTIMIZATION FOR NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

The optimization of large language models (LLMs) remains a critical challenge, particularly as model scaling exacerbates sensitivity to algorithmic imprecision and training instability. Recent advances in optimizers improve convergence efficiency through momentum orthogonalization but suffer from two key robustness limitations: dimensional fragility in orthogonalization precision and vulnerability to outlier-induced noise. To address these robustness challenges, we introduce FIRMO, an optimizer that enhances training stability through dual robustness mechanisms. First, we develop a dimension-robust orthogonalization scheme that calibrates the full Newton–Schulz composition by matrix shape, achieving higher fidelity than Muon across the evaluated shapes under the same five-step budget. Second, we introduce an optimization-robust framework grounded in proximal optimization that adaptively suppresses extreme input magnitudes while preserving entries within the threshold and retaining matrix-wise orthogonalization. Experiments on a 1B-parameter LLM demonstrate faster convergence in training steps and lower final loss than Muon in 10B-token runs. After 100B-token pre-training, FIRMO achieves 60.12% average zero-shot accuracy across nine tasks, versus 59.59% for Muon and 59.05% for AdamW. Our work establishes a unified framework for robust and precise matrix-aware optimization by jointly exploiting matrix-shape structure and adaptive input-scale control.

1 INTRODUCTION

Pre-training large language models requires optimizers that combine efficient computation with reliable matrix updates (Brown et al., 2020; OpenAI, 2023; Chen et al., 2025). AdamW (Kingma & Ba, 2014; Loshchilov & Hutter, 2017) adapts updates coordinatewise, whereas Muon (Jordan et al., 2024; Liu et al., 2025) applies approximate polar orthogonalization to matrix-valued momentum. This matrix-wise approach motivates jointly considering finite-step approximation and control of the input supplied to orthogonalization.

FIRMO exploits two complementary sources of structure in fixed-budget orthogonalization: the geometry induced by matrix shape and the scale distribution of the pre-orthogonalization input. With a fixed Newton–Schulz budget, approximation quality varies across the tested matrix shapes, creating an opportunity to align the polynomial with matrix geometry. At the same time, the scale distribution of the pre-orthogonalization input provides a natural signal for adaptive entrywise control. These observations motivate a unified design in which the finite-step approximation is shape-conditioned and the input scale is controlled before the matrix transformation.

To exploit these two forms of structure, we introduce **FIRMO** (**F**inite-step **I**nput-**R**obust **M**atrix **O**rthogonalization), a matrix-aware optimizer that retains matrix-wise orthogonalization while adapting its finite-step polynomial to matrix shape and its input-scale control to each current input. During an offline calibration stage, FIRMO constructs a shape-indexed Newton–Schulz coefficient table under a fixed five-step budget. During model training, it computes a percentile threshold for each pre-orthogonalization input, clips its entries, and applies the calibrated polynomial. The resulting design aligns finite-step approximation with matrix geometry and controls the scale of the matrix entering orthogonalization. We provide an exact finite-step spectral characterization and a proximal characterization of the clipping operator.

The main contributions of this work are summarized as follows:

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.

- We develop shape-conditioned coefficient calibration for orthogonalization under a fixed five-step Newton–Schulz budget. The coefficients are calibrated offline for each matrix shape and remain fixed during training, reducing the approximation error on the evaluated shapes.
- We combine this calibrated approximation with percentile-based entrywise clipping before orthogonalization, with a threshold recomputed for each input matrix. An exact finite-step spectral characterization and a proximal interpretation clarify the complementary roles of the two components.
- We evaluate FIRMO in 1B-parameter LLM pre-training, with 10B-token component comparisons and zero-shot evaluation after training on 100B tokens. Across nine benchmarks, FIRMO achieves a mean accuracy of 60.12%, compared with 59.59% for Muon and 59.05% for AdamW, and exceeds Muon on seven of the nine tasks.

2 Related Work

In this section, we provide a brief overview of related work in the field of optimizers, as well as recent advances.

2.1 Matrix-Aware Optimizers

The training of deep neural networks has historically relied on stochastic first-order methods (Robbins & Monro, 1951). Standard optimizers, such as SGD with momentum (Polyak, 1964) and AdamW (Kingma & Ba, 2014; Loshchilov & Hutter, 2017), treat model parameters as independent vectors. While computationally efficient, these coordinate-wise updates overlook the rich structural correlations inherent in weight matrices. To address this, second-order methods like K-FAC (Martens & Grosse, 2015) and Shampoo (Gupta et al., 2018; Anil et al., 2020) leverage Kronecker-factored preconditioners to capture parameter geometry. However, despite their theoretical convergence advantages, these methods often incur prohibitive computational and memory overheads in large-scale settings.

Bridging this gap, the Muon optimizer (Jordan et al., 2024) regulates update geometry through matrix orthogonalization rather than explicit curvature approximation. Specifically, Muon employs a Newton-Schulz iteration Bernstein & Newhouse (2024); Higham (2008); Guo & Higham (2006) to orthogonalize the momentum matrix, which theoretically corresponds to performing steepest descent under the spectral norm (Li & Hong, 2025; Kovalev, 2025). Muon implements matrix-wise orthogonalization using a fixed number of dense matrix multiplications. FIRMO specializes this five-step computation to matrix shape through offline coefficient calibration, with no additional NS matrix multiplications. Training reuses the frozen coefficient table and applies per-input quantile computation and entrywise clipping.

2.2 Recent Advances in Muon Variants

Following Muon's success, several variants have emerged to extend its capabilities across different dimensions:

Efficiency and Scalability. To mitigate communication overhead in distributed settings, Dion (Ahn et al., 2025) replaces the dense Newton-Schulz iteration with amortized power iteration. For computational efficiency, LiMuon (Huang et al., 2025) leverages randomized SVD, while Drop-Muon (Gruntkowska et al., 2025) explores randomized layer subsampling to reduce the update frequency.

Adaptivity and Precision. While Muon lacks element-wise adaptivity, recent works attempt to reintegrate it. AdaGO (Zhang et al., 2025) combines orthogonal directions with AdaGrad-style step sizes (Duchi et al., 2011). AdaMuon (Si et al., 2025) incorporates a second-moment estimator, employing sign-based orthogonalization to enforce stability. CANS (Grishina et al., 2025) designs Chebyshev-type polynomials over spectral intervals, while Polar Express (Amsel et al., 2025) constructs iteration-specific minimax polynomials. FIRMO minimizes the error of the full five-step composition over shape-conditioned calibration spectra and combines the calibrated map with input-dependent percentile control. Each matrix shape uses one offline-calibrated coefficient triplet shared across all five iterations.

3 APPROACH

FIRMO retains matrix-wise orthogonalization and introduces shape-conditioned calibration and input-dependent scale control. It uses a shape-indexed coefficient table calibrated offline and a percentile-based clipping operator recomputed for each input matrix. We first describe the baseline orthogonalization interface, then present shape-conditioned calibration and adaptive input-scale control.

3.1 PRELIMINARIES

The orthogonalization-based optimizers, *e.g.*, Muon Jordan et al. (2024), address the optimization of neural network parameters that exhibit a matrix structure. During each iteration t, the algorithm updates the weight matrix θ_{t-1} using the current momentum μ, learning rate η_t, and objective function $\mathcal{L}$. The optimization procedure is defined by the following steps:

$$\begin{aligned} M_t &= \mu M_{t-1} + \nabla \mathcal{L}(\theta_{t-1}) \\ M'_t &= \text{Newton-Schulz}(M_t) \\ \theta_t &= \theta_{t-1} - \eta_t M'_t \end{aligned} \tag{1}$$

In this formulation, M_t represents the gradient momentum at step t, initialized as a zero matrix when $t = 0$. The Newton–Schulz (NS) routine Bernstein & Newhouse (2024) approximates the polar factor of the matrix supplied to the orthogonalization interface. For a full-rank thin SVD $M_t = U\Sigma V^{\mathsf{T}}$, the corresponding polar-factor update is UV^{T}. This matrix-aware transformation preserves the coherent geometry of Muon updates while avoiding an elementwise treatment of the weight matrix. In FIRMO, we denote the actual pre-orthogonalization input at this interface by X_t; this notation accommodates the outer momentum, scaling, and implementation details used by the training pipeline.

The iterative process begins with a normalized input $X_{t,0}$ derived from the matrix supplied to the orthogonalization routine. At each inner step k, the matrix $X_{t,k}$ is updated according to the recurrence relation:

$$\begin{aligned} X_{t,k} =& aX_{t,k-1} + bX_{t,k-1}(X_{t,k-1}^T X_{t,k-1}) \\ &+ cX_{t,k-1}(X_{t,k-1}^T X_{t,k-1})^2 \end{aligned} \tag{2}$$

After a fixed budget of $K = 5$ iterations, the resulting matrix is used as the orthogonalized update. The Muon baseline uses a globally shared coefficient triplet, $a = 3.4445$, $b = -4.7750$, and $c = 2.0315$. FIRMO retains the same five-step NS iteration budget and replaces the global coefficient triplet with a frozen shape-indexed table.

3.2 SHAPE-CONDITIONED NEWTON–SCHULZ CALIBRATION

3.2.1 EXPLOITING MATRIX-SHAPE STRUCTURE

The five-step NS operator acts on matrices with heterogeneous shapes. A globally shared coefficient triplet cannot be expected to align equally well with every matrix geometry. As shown in Table 1, the reported approximation error varies across the evaluated shapes, while shape-conditioned coefficients substantially reduce the error for the same five-step budget. For example, at $m = n = 2048$, the reported MSE decreases from 0.0499 to 0.0352. These results motivate a shape-conditioned calibration strategy that assigns each architectural matrix shape a coefficient triplet matched to its finite-step orthogonalization behavior.

We therefore construct a shape-conditioned Newton–Schulz operator, denoted AdaNewton in the experiments, with specialized coefficients $\{a^{(m,n)}, b^{(m,n)}, c^{(m,n)}\}$ for each matrix shape (m, n). The resulting coefficient table is frozen for use in the reported training runs and accessed by shape lookup, preserving a compact and deterministic online update rule.

Formally, for a matrix $X_{t,k-1} \in \mathbb{R}^{m\times n}$, our shape-conditioned update rule at step t becomes:

$$\begin{aligned} X_{t,k} =& a^{(m,n)} X_{t,k-1} + b^{(m,n)} X_{t,k-1}(X_{t,k-1}^T X_{t,k-1}) \\ &+ c^{(m,n)} X_{t,k-1}(X_{t,k-1}^T X_{t,k-1})^2 \end{aligned} \tag{3}$$

Table 1: Five-step orthogonalization across matrix shapes. The Muon baseline uses $(a, b, c) = (3.4445, -4.7750, 2.0315)$; FIRMO uses the shape-specific coefficients below. Lower MSE is better.

Matrix shape		FIRMO coefficients			MSE ↓	
n	m	a	b	c	Muon	FIRMO
2048	2048	3.3334	-4.2591	1.7791	0.0499	**0.0352**
4096	4096	3.3732	-4.6134	2.0576	0.0637	**0.0470**
8192	8192	3.3886	-4.9026	2.3105	0.0761	**0.0587**
2048	3072	2.9091	-3.8108	1.8600	0.0338	**0.0024**
2048	4096	2.7739	-3.4911	1.6992	0.0362	**0.0010**
2048	8192	2.5925	-3.0010	1.4138	0.0340	**0.0003**
2048	16384	2.7045	-3.2335	1.5336	0.0425	**0.0003**

The table adapts the finite-step polynomial to each matrix shape without introducing trainable optimizer state. FIRMO combines Muon's matrix-aware update geometry with shape-conditioned finite-step calibration and adaptive input-scale control. It calibrates a shape-indexed Newton–Schulz coefficient table offline, recomputes a percentile threshold for each training input, and applies clipping followed by five-step orthogonalization. The exact finite-step spectral characterization explains the role of the calibrated coefficients, while the proximal characterization provides a principled view of clipping. Across the reported LLM pre-training, component, and vision experiments, these mechanisms provide complementary improvements in orthogonalization fidelity and downstream training behavior.

3.2.2 Exact Finite-Step Spectral Characterization

The finite-step behavior reduces to a scalar polynomial acting on the normalized input spectrum.

For a fixed matrix shape $s = (m, n)$, let $\widehat{\boldsymbol{\alpha}}_s = (a_s, b_s, c_s)$ denote the offline-calibrated triplet reused at every inner iteration. Define

$$g_s(z) = a_s z + b_s z^3 + c_s z^5, \qquad h_{s,K} = \underbrace{g_s \circ \cdots \circ g_s}_{K \text{ compositions}}, \qquad K = 5.$$

To make the calibration target explicit, let $\mathcal{C}_s = \{X_{s,j}\}_{j=1}^{N_s}$ denote the calibration matrices assigned to shape s, and let $\{\sigma_{s,j,i}\}_{i=1}^{r_s}$ be the singular values of the normalized calibration matrices, where $r_s = \min(m, n)$. We specify the empirical five-step spectral objective

$$\widehat{\mathcal{J}}_s(a_s, b_s, c_s) = \frac{1}{N_s r_s} \sum_{j=1}^{N_s} \sum_{i=1}^{r_s} \left[h_{s,K}(\sigma_{s,j,i}) - 1\right]^2. \tag{4}$$

For each calibration strategy, $\mathcal{C}_s$ contains the corresponding empirical/random mixture; FIRMO obtains the coefficient triplet by minimizing this objective and freezes the resulting shape-indexed table for training.

Let $Z_{t,0} \in \mathbb{R}^{m \times n}$ be a normalized, full-rank input to the NS routine with thin SVD

$$Z_{t,0} = U \operatorname{diag}(\sigma_1, \ldots, \sigma_r) V^{\mathsf{T}}, \qquad r = \min(m, n), \quad \sigma_i > 0.$$

The associated polar factor is $Q_t = UV^{\mathsf{T}}$. Because the NS recurrence is a matrix polynomial in $Z_{t,0}^{\mathsf{T}} Z_{t,0}$, the singular-vector factors remain fixed and, in exact arithmetic,

$$Z_{t,K} = U \operatorname{diag}\big(h_{s,K}(\sigma_1), \ldots, h_{s,K}(\sigma_r)\big) V^{\mathsf{T}}.$$

Consequently, the finite-step approximation error is exactly

$$\begin{aligned} \mathcal{L}_{\mathrm{NS}}\big(\widehat{\boldsymbol{\alpha}}_s; Z_{t,0}\big) &= \|Z_{t,K} - Q_t\|_F^2 \\ &= \sum_{i=1}^{r} \big(h_{s,K}(\sigma_i) - 1\big)^2. \end{aligned} \tag{5}$$

This identity connects the coefficient table to five-step orthogonalization fidelity; its training effects are evaluated in Sections 4.2 and 4.3.

Calibration and polar-update alignment. Under the assumptions above, the five-step output satisfies

$$\langle Z_{t,0}, Z_{t,5}\rangle \geq \|Z_{t,0}\|_* - \|Z_{t,0}\|_F\sqrt{\mathcal{L}_{\mathrm{NS}}(\widehat{\boldsymbol{\alpha}}_s; Z_{t,0})}.$$

Indeed, $\langle Z_{t,0}, Q_t\rangle = \|Z_{t,0}\|_*$, and Cauchy–Schwarz bounds $\langle Z_{t,0}, Z_{t,5} - Q_t\rangle$ from below by $-\|Z_{t,0}\|_F\|Z_{t,5} - Q_t\|_F$. Thus, the finite-step approximation error controls a bound on the linear-alignment shortfall relative to exact polar orthogonalization.

3.3 Robust Orthogonalization via Input-Scale Control

The scale distribution of the matrix entering orthogonalization is an additional source of structure that can be controlled online. Large-magnitude entries can dominate the normalization and polynomial transformation applied by a finite-step NS routine. Figure 1 provides a conceptual illustration of the heavy-tailed input patterns motivating FIRMO's adaptive input-scale control.

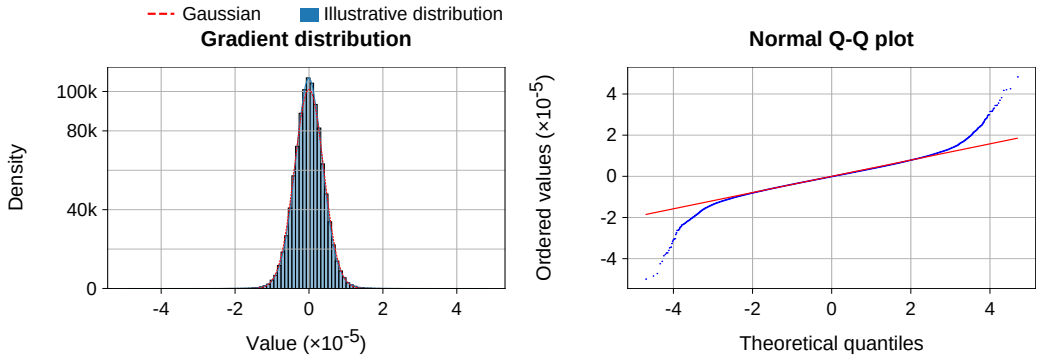


Figure 1: Conceptual illustration of the input-scale heterogeneity motivating FIRMO. **Left:** histogram with a Gaussian reference. **Right:** normal Q–Q plot illustrating tail departures from the reference. These patterns motivate adaptive control of extreme input magnitudes before orthogonalization.

FIRMO therefore introduces an input-control operator at the orthogonalization interface. For each input matrix, a percentile-derived threshold adapts to its current scale, and entrywise clipping limits the magnitude presented to the matrix transformation while retaining the matrix-wise orthogonalization framework.

3.3.1 Clipping and Its Proximal Interpretation

Let $X_t \in \mathbb{R}^{m\times n}$ denote the matrix supplied to the orthogonalization routine immediately before orthogonalization. For a chosen percentile $p \in [0,1]$, FIRMO computes separately for each input matrix

$$\varepsilon_t = \mathrm{Quantile}\big(\{|(X_t)_{ij}|\}_{i,j}, p\big), \qquad B_t = \mathrm{clip}(X_t, -\varepsilon_t, \varepsilon_t). \tag{6}$$

The clipped matrix B_t is the retained input to orthogonalization. Define the removed excess as $O_t = X_t - B_t$, so that

$$X_t = B_t + O_t. \tag{7}$$

For the computed threshold ε_t, treated as fixed in the following subproblem, O_t has the proximal characterization

$$O_t = \arg\min_O \left\{\frac{1}{2}\|X_t - O\|_F^2 + \varepsilon_t\|O\|_{1,\mathrm{entry}}\right\}, \qquad \|O\|_{1,\mathrm{entry}} = \sum_{i,j}|O_{ij}|. \tag{8}$$

This strictly convex problem separates entrywise. Its unique solution is the soft-thresholding operator Tibshirani (1996); Donoho (1995):

$$(O_t)_{ij} = [\mathcal{T}_{\varepsilon_t}(X_t)]_{ij} = \mathrm{sign}((X_t)_{ij})\max\{|(X_t)_{ij}| - \varepsilon_t, 0\}. \tag{9}$$

Algorithm 1 FIRMO Orthogonalization Procedure

Require: Separate 0.5B-token calibration sample, percentile p, $K = 5$

Offline: group calibration matrices into $\{\mathcal{C}_s\}$ by shape.
For each s, form $\widehat{\mathcal{J}}_s$ from normalized spectra, minimize it to obtain (a_s, b_s, c_s), and freeze the table.
At training step t, for each pre-orthogonalization input X_t:
$s \leftarrow \mathrm{shape}(X_t)$
$\varepsilon_t \leftarrow \mathrm{Quantile}(\{|(X_t)_{ij}|\}_{i,j}, p)$
$B_t \leftarrow \mathrm{clip}(X_t, -\varepsilon_t, \varepsilon_t)$
$Z_{t,0} \leftarrow \mathrm{Normalize}(B_t)$
$(a_s, b_s, c_s) \leftarrow \widehat{\boldsymbol{\alpha}}_s$
for $k = 0, \ldots, K-1$ **do**
    $A_{t,k} \leftarrow Z_{t,k}^{\mathsf{T}} Z_{t,k}$
    $Z_{t,k+1} \leftarrow a_s Z_{t,k} + b_s Z_{t,k} A_{t,k} + c_s Z_{t,k} A_{t,k}^2$
end for
Return $Z_{t,K}$ to the parameter-update routine.

Consequently,

$$B_t = X_t - \mathcal{T}_{\varepsilon_t}(X_t) = \mathrm{clip}(X_t, -\varepsilon_t, \varepsilon_t). \tag{10}$$

Thus, soft-thresholding gives a variational characterization of the removed excess, whereas entrywise clipping describes the matrix passed to the Newton–Schulz iteration. FIRMO combines this principled input-control operator with the shape-conditioned polynomial described above.

Fixed-budget deployment. FIRMO preserves Muon's five-step Newton–Schulz budget: shape conditioning changes only coefficient lookup, with no extra inner iteration, trainable optimizer state, or Newton–Schulz multiplication. Training adds a per-input percentile selection and entrywise clipping before normalization; the calibrated shape-indexed table is frozen, so the update remains compact and deterministic.

4 Experiments

4.1 Implementation Details

Experiments Setup To validate our approach, we conduct pretraining using the FineWeb-Edu dataset Penedo et al. (2024), utilizing a 10-billion-token subset for ablation studies and a 100-billion-token sample for the main experiment. Both subsets are available at HuggingFaceFW (2024). To systematically evaluate the proposed optimization enhancements, we train a 1B Transformer Dubey et al. (2024); Rang et al. (2025). All models are trained on distributed clusters of Ascend NPUs. The training infrastructure leverages high-speed interconnects to enable efficient data and model parallelism, ensuring scalable training across multiple nodes. An attention-mask-reset strategy Chen et al. (2025) is also used to prevent self-attention between different documents within a sequence.

Regarding the hyperparameters, all models are pretrained for a single epoch. We employ a cosine learning rate schedule that decays to 10% of the peak learning-rate, following a warm-up phase of 2,000 steps. Specifically, for the 10B-token ablation experiments, the peak learning rate is set to 8×10^{-4} with a global batch size of 0.4M. The pre-training sequence length is set to 4096. For the 100B-token main experiments, the learning rate is increased to 1.6×10^{-3}, and the global batch size is set to 1M. We adopt the default Muon hyperparameters from Jordan et al. (2024) and apply a 0.2 scaling factor following Liu et al. (2025) to align the update RMS with AdamW.

Evaluation Benchmark Setup We evaluate our method on a comprehensive set of Academic benchmarks: HellaSwag Zellers et al. (2019), ARC-easy (ARC-e) and ARC-challenge (ARC-c) Clark et al. (2018), BoolQ Clark et al. (2019), PIQA Bisk et al. (2020), SciQ Welbl et al. (2017), WINO Sakaguchi et al. (2019), OBQA Mihaylov et al. (2018), and WSC Levesque et al. (2012). We utilize the lm-evaluation-harness framework Sutawika et al. (2024) for all evaluations, and we evaluate all tasks in zero-shot.

4.2 Validation on Real Training Dynamics

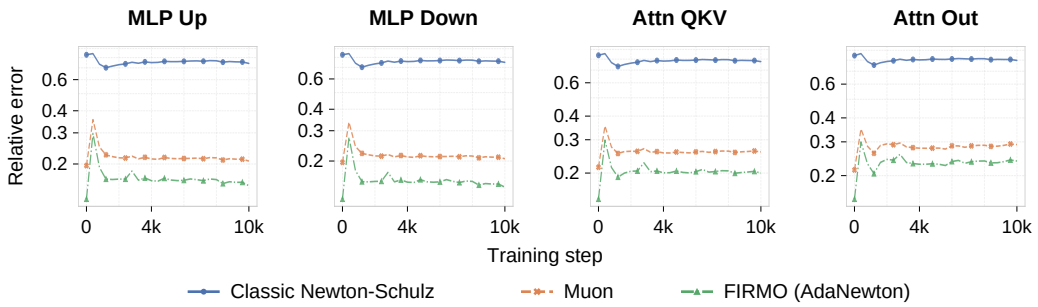


Figure 2: **Orthogonalization precision relative to ground-truth SVD.** The plot tracks the Relative Error averaged over all optimized parameters (Attention QKV/O and MLP Up/Down projections). Under a fixed 5 iterations, FIRMO maintains lower approximation error compared to the Muon baseline and Classic Newton-Schulz. This indicates that shape-specific coefficients provide superior fidelity across varying matrix dimensions.

While Table 1 evaluates shape-specific coefficients on controlled calibration inputs, we also test transfer to input gradients G_t sampled from the first 10k pre-training steps; these evaluation inputs are separate from the calibration sample.

In this experiment, we compare three distinct orthogonalization strategies, each restricted to a strict budget of 5 iterations. We evaluate the Muon Baseline Jordan et al. (2024) with its general-purpose coefficients ($a = 3.4445, b = -4.7750, c = 2.0315$), as well as the Classic Newton-Schulz (Quintic) which uses coefficients $(1.875, -1.25, 0.375)$. We contrast them with FIRMO (AdaNewton), whose frozen table is obtained from the separate calibration sample.

To quantify approximation fidelity, we report the Relative Error ($\|\hat{O} - O_{\text{SVD}}\|_F / \|O_{\text{SVD}}\|_F$) relative to the ground-truth SVD. The reported error is averaged across all parameter types targeted by the optimizer (Attention Query/Output and MLP Up/Down projections) at each step.

Figure 2 illustrates the evolution of this metric. FIRMO consistently maintains a lower relative error compared to the baselines throughout the training trajectory. While the Muon baseline exhibits a distinguishable error floor, likely due to the sub-optimality of fixed coefficients for specific aspect ratios, FIRMO achieves a closer approximation to the ground truth. These results suggest that incorporating dimension-aware coefficients mitigates the precision loss often associated with fixed-coefficient Newton-Schulz iterations on diverse matrix shapes.

4.3 LLM Pre-training Analysis

We evaluate the pre-training performance of the proposed FIRMO optimizer on the 1B Transformer over a 10B-token trajectory. Figure 3 visualizes the training loss. We compare the Muon baseline against two configurations: *FIRMO-SoftThresh* (outlier suppression only) and the full FIRMO optimizer.

As observed, both FIRMO variants achieve consistently lower training loss compared to the Muon baseline. The FIRMO (SoftThresh) variant improves convergence over standard Muon, suggesting that suppressing gradient outliers aids in stabilizing the optimization process. Furthermore, the full FIRMO optimizer yields the lowest loss, indicating that dimension-adaptive orthogonalization provides additive benefits to outlier suppression. Ultimately, FIRMO reaches a final training loss of 2.5407, surpassing the Muon baseline by 0.01.

4.4 Benchmark Evaluation Results

To assess model generalization, we further train the 1B model on 100B tokens and evaluate the trained 1B models across a diverse set of common academic benchmarks. Table 2 summarizes the zero-shot performance. FIRMO achieves competitive or superior performance across the evaluated

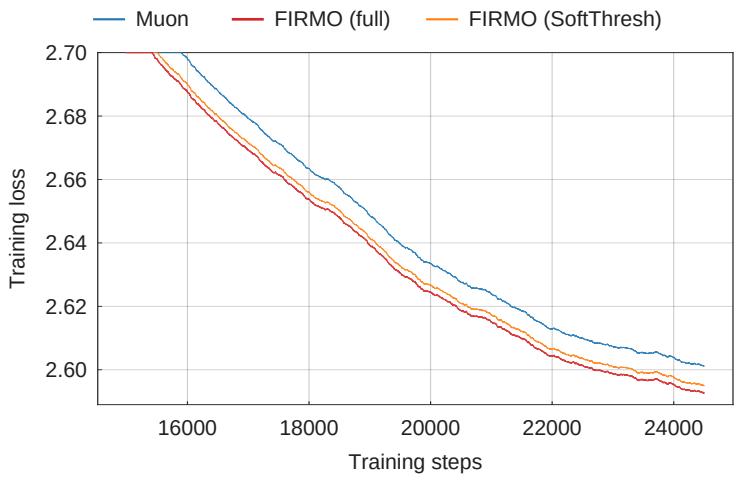


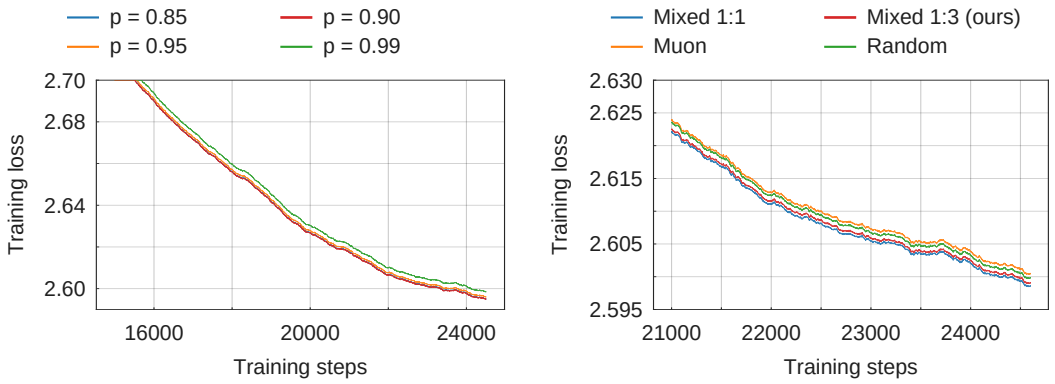
Figure 3: Training loss curves for 10B tokens. FIRMO variants demonstrate faster convergence and lower final loss compared to Muon baseline, with full FIRMO achieving the best performance.

Table 2: Zero-shot accuracy (%) after 100B-token pre-training. Bold denotes the best result in each column.

Method	HellaSwag	BoolQ	PIQA	ARC-e	ARC-c	OBQA	SciQ	Wino	WSC	Avg.
AdamW	44.24	**62.60**	72.69	71.63	**37.80**	27.20	89.80	58.09	67.40	59.05
Muon	44.83	61.16	73.07	**74.12**	37.12	29.80	89.50	59.67	67.03	59.59
FIRMO	**45.37**	62.08	**73.12**	72.14	36.86	**31.20**	**90.40**	**60.30**	**69.60**	**60.12**

tasks. These results confirm that FIRMO enhancements not only accelerate training convergence but also yield higher-quality final language models.

4.5 Ablation Studies



(a) Quantile percentile p.

(b) Coefficient calibration data.

Figure 4: Ablation studies on 10B-token training. **(a)** Percentile $p = 0.90$ balances suppression and signal preservation. **(b)** Mixed (1:1) gives lower loss here, whereas Mixed (1:3) is selected for its balance of convergence and robustness in the broader configurations discussed in the text.

Outlier Suppression Threshold. Instead of a fixed threshold scalar, we employ a dynamic, percentile-based threshold $\varepsilon_t = \text{Quantile}(|X_t|, p)$ to adapt to the evolving input scale. We investigate the sensitivity of convergence to varying percentiles $p \in \{0.85, 0.90, 0.95, 0.99\}$. As illustrated in

Figure 4(a), the results reveal a trade-off between outlier suppression and signal preservation. A conservative threshold ($p = 0.99$) results in under-suppression, failing to filter the heavy tail of the noise distribution and leading to sub-optimal stability. Conversely, an overly aggressive threshold ($p = 0.85$) causes excessive truncation. We identify $p = 0.90$ as the optimal equilibrium for LLM pre-training tasks, effectively isolating outliers while retaining the structural integrity of the gradients.

Spectral Calibration Strategy. We construct the shape-indexed coefficients from the separate 0.5B-token calibration sample and compare three input mixtures: Random Calibration (random Gaussian matrices only), Mixed (1:1), and Mixed (1:3), where the ratios are calibration to random matrices.

Figure 4(b) illustrates the training trajectories under these configurations. While the Mixed (1:1) strategy yields the lowest loss in this specific setting, broader evaluations reveal its vulnerability to instability (e.g., loss spikes) when scaled to larger models or integrated with the Soft-Thresholding mechanism. Conversely, the Random Calibration baseline offers limited convergence acceleration. Consequently, we adopt the Mixed (1:3) strategy to balance accelerated convergence with stability across the considered configurations.

4.6 Generalization to Vision Tasks

Table 3: CIFAR-10 top-1 test accuracy (6.3M-parameter ViT, trained from scratch). Muon LR $= 0.02$, AdamW LR $= 0.001$, weight decay $= 5 \times 10^{-5}$.

Method	Muon	FIRMO ($p = 0.95$)	FIRMO ($p = 0.90$)	FIRMO ($p = 0.85$)
Accuracy (%)	84.67	85.75	86.58	**88.44**

To evaluate the generalization capabilities of FIRMO beyond language modeling, we conducted image classification experiments by training a lightweight Vision Transformer ($\approx$ 6.3M parameters) on CIFAR-10 (Krizhevsky, 2009) from scratch. We adopt a compact architecture adapted from omihub777 (2021), which processes images via 4×4 patches. In this experiment, we explicitly isolate the efficacy of the Soft-Thresholding mechanism against the Muon baseline. Models were trained for 100 epochs. Following standard protocols, all 2D weight matrices were optimized via Muon or FIRMO, while 1D parameters (biases, norms) and embeddings and class-tokens were optimized using AdamW. The results in Table 3 show that FIRMO consistently outperforms the baseline. The improvement is most significant for the quantile percentile hyperparameter of 0.85, where the baseline achieves only 84.67% accuracy. These results confirm that the soft-thresholding mechanism mitigates gradient noise, thereby enhancing generalization even in non-language modalities.

5 Conclusion

FIRMO combines Muon's matrix-aware update geometry with shape-conditioned finite-step calibration and adaptive input-scale control. It calibrates a shape-indexed Newton–Schulz coefficient table offline, recomputes a percentile threshold for each training input, and applies clipping followed by five-step orthogonalization. The exact finite-step spectral characterization explains the role of the calibrated coefficients, while the proximal characterization provides a principled view of clipping. Across the reported LLM pre-training, component, and vision experiments, these mechanisms provide complementary improvements in orthogonalization fidelity and downstream training behavior. The resulting design turns the globally shared five-step map into a shape-conditioned operator without increasing the orthogonalization budget: shape conditioning targets geometry-dependent finite-step error, while input-scale control clips extreme entries before the matrix transformation. Because the coefficient table is frozen and indexed only by shape, the online rule remains deterministic and compact, preserving Muon's matrix-wise update geometry while adding no trainable optimizer state.

AI USE STATEMENT

Generative AI tools assisted with language polishing and editing of the readability of the manuscript.

REFERENCES

Kwangjun Ahn, Byron Xu, Natalie Abreu, Ying Fan, Gagik Magakyan, Pratyusha Sharma, Zheng Zhan, and John Langford. Dion: Distributed orthonormalized updates. *arXiv preprint arXiv:2504.05295*, 2025. doi: 10.48550/arXiv.2504.05295.

Noah Amsel, David Persson, Christopher Musco, and Robert M. Gower. The Polar Express: Optimal matrix sign methods and their application to the Muon algorithm. *arXiv preprint arXiv:2505.16932*, 2025. doi: 10.48550/arXiv.2505.16932. URL https://arxiv.org/abs/2505.16932.

Rohan Anil, Vineet Gupta, Tomer Koren, Kevin Regan, and Yoram Singer. Scalable second order optimization for deep learning. *arXiv preprint arXiv:2002.09018*, 2020. doi: 10.48550/arXiv.2002.09018.

Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. *arXiv preprint arXiv:2409.20325*, 2024. doi: 10.48550/arXiv.2409.20325.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020.

Hanting Chen, Yasheng Wang, Kai Han, Dong Li, Lin Li, Zhenni Bi, Jinpeng Li, Haoyu Wang, Fei Mi, Mingjian Zhu, et al. Pangu Embedded: An efficient dual-system LLM reasoner with metacognition. *arXiv preprint arXiv:2505.22375*, 2025. doi: 10.48550/arXiv.2505.22375.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. doi: 10.48550/arXiv.1803.05457.

David L Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3): 613–627, 1995. doi: 10.1109/18.382009.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783v1*, 2024. doi: 10.48550/arXiv.2407.21783. URL https://arxiv.org/abs/2407.21783v1.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011. URL https://jmlr.org/papers/v12/duchi11a.html.

Ekaterina Grishina, Matvey Smirnov, and Maxim Rakhuba. Accelerating Newton-Schulz iteration for orthogonalization via Chebyshev-type polynomials. *arXiv preprint arXiv:2506.10935*, 2025. doi: 10.48550/arXiv.2506.10935.

Kaja Gruntkowska, Yassine Maziane, Zheng Qu, and Peter Richtárik. Drop-Muon: Update less, converge faster. *arXiv preprint arXiv:2510.02239*, 2025. doi: 10.48550/arXiv.2510.02239.

Chun-Hua Guo and Nicholas J Higham. A Schur–Newton method for the matrix pth root and its inverse. *SIAM Journal on Matrix Analysis and Applications*, 28(3):788–804, 2006. doi: 10.1137/050643374.

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

Nicholas J Higham. *Functions of matrices: theory and computation*. SIAM, 2008. doi: 10.1137/1.9780898717778.

Feihu Huang, Yuning Luo, and Songcan Chen. LiMuon: Light and fast Muon optimizer for large models. *arXiv preprint arXiv:2509.14562*, 2025. doi: 10.48550/arXiv.2509.14562.

HuggingFaceFW. FineWeb-Edu (revision 22b0aca), 2024. URL https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu.

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

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. doi: 10.48550/arXiv.1412.6980.

Dmitry Kovalev. Understanding gradient orthogonalization for deep learning via non-Euclidean trust-region optimization. *arXiv preprint arXiv:2503.12645*, 2025. doi: 10.48550/arXiv.2503.12645.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://cave.cs.toronto.edu/kriz/learning-features-2009-TR.pdf.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The Winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, pp. 552–561. AAAI Press, 2012. ISBN 9781577355601. URL https://cdn.aaai.org/ocs/4492/4492-21843-1-PB.pdf.

Jiaxiang Li and Mingyi Hong. A note on the convergence of Muon and further. *arXiv preprint arXiv:2502.02900v1*, 2025. doi: 10.48550/arXiv.2502.02900. URL https://arxiv.org/abs/2502.02900v1.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for LLM training. *arXiv preprint arXiv:2502.16982*, 2025. doi: 10.48550/arXiv.2502.16982.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. doi: 10.48550/arXiv.1711.05101.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

omihub777. ViT-CIFAR: Vision transformer for CIFAR-10/100 on PyTorch. GitHub repository, 2021. URL https://github.com/omihub777/ViT-CIFAR. Accessed: 2026-09-24.

OpenAI. GPT-4 technical report. Technical report, OpenAI, 2023. URL https://cdn.openai.com/papers/gpt-4.pdf.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, pp. 30811–30849, 2024. doi: 10.52202/079017-0970.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964. doi: 10.1016/0041-5553(64)90137-5.

Miao Rang, Zhenni Bi, Hang Zhou, Hanting Chen, An Xiao, Tianyu Guo, Kai Han, Xinghao Chen, and Yunhe Wang. Revealing the power of post-training for small language models via knowledge distillation. *arXiv preprint arXiv:2509.26497*, 2025. doi: 10.48550/arXiv.2509.26497.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951. doi: 10.1214/aoms/1177729586.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019. doi: 10.48550/arXiv.1907.10641.

Chongjie Si, Debing Zhang, and Wei Shen. AdaMuon: Adaptive Muon optimizer. *arXiv preprint arXiv:2507.11005*, 2025. doi: 10.48550/arXiv.2507.11005.

Lintang Sutawika, Hailey Schoelkopf, Leo Gao, Baber Abbasi, Stella Biderman, Jonathan Tow, ben fattori, Charles Lovering, farzanehnakhaee70, Jason Phang, Anish Thite, Fazz, Aflah, Niklas Muennighoff, Thomas Wang, sdtblck, nopperl, gakada, tttyuntian, researcher2, Julen Etxaniz, Chris, Hanwool Albert Lee, Zdeněk Kasner, Khalid, LSinev, Jeffrey Hsu, Anjor Kanekar, KonradSzafer, and AndyZwei. EleutherAI/lm-evaluation-harness: v0.4.3, July 2024. URL https://zenodo.org/records/12608602.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. doi: 10.1111/j.2517-6161.1996.tb02080.x.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 94–106. Association for Computational Linguistics, 2017. doi: 10.18653/v1/W17-4413. URL https://aclanthology.org/W17-4413/.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, jul 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Minxin Zhang, Yuxuan Liu, and Hayden Schaeffer. AdaGrad meets Muon: Adaptive stepsizes for orthogonal updates. *arXiv preprint arXiv:2509.02981*, 2025. doi: 10.48550/arXiv.2509.02981.