ACCELERATING MUON WITH SPECTRAL RESIDUAL CORRECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Muon constructs matrix updates by applying a matrix-sign transformation to momentum. A common route to accelerating Muon is to modify the gradient or momentum before this transformation. We explore a complementary direction: refining the parameter step while retaining Muon's momentum recursion. Derived from the parameter subproblem of the ADMM-Inspired Momentum framework, AIMuon uses the gradient–momentum residual to correct the tentative spectral step. The residual undergoes a separate matrix-sign transformation, while spectral–nuclear duality determines the relative weighting of the momentum and residual directions. In practice, both directions are approximated with Newton–Schulz iterations and their combination is rescaled to match Muon's update magnitude. We establish a stochastic stationarity bound for the corresponding exact iteration. On GPT-2-small and GPT-2-medium pretraining, AIMuon reaches Muon's final validation loss with 12.8% and 7.7% fewer optimization steps, respectively, while achieving lower final validation loss than both Muon and MONA. Ablation studies further support the roles of the residual correction, separate transformation, adaptive nuclear-norm weighting, and final normalization. These results show that parameter-step refinement can complement momentum construction in matrix-sign optimization.

1 INTRODUCTION

Improving optimization efficiency is an important goal in neural network training. Muon (Jordan et al., 2024) has recently emerged as an effective optimizer for matrix parameters in large language model training (Liu et al., 2025; Qiu et al., 2026). Unlike coordinate-wise adaptive methods, Muon approximately orthogonalizes the momentum matrix using Newton–Schulz iterations (Schulz, 1933), producing a matrix-sign-like update that acts on its singular structure. Its strong empirical performance naturally raises the question of how Muon can be further accelerated.

One established route to acceleration is to improve how gradient information is accumulated into momentum. Adan (Xie et al., 2024) incorporates gradient differences into both first- and second-order moment estimates through a reformulation of Nesterov acceleration. ALTO (Zhao et al., 2025) uses gradient differences as a low-cost curvature proxy and applies their EMA to correct the gradient entering the momentum estimates. MONA (Li et al., 2026a) incorporates related gradient-difference information into Muon's momentum before applying the matrix-sign transformation. These methods illustrate how gradient changes can improve the signal accumulated by momentum.

For Muon, the effect of a correction depends on whether it is introduced before or after the matrix-sign transformation. With a linear transformation, adding a correction before the transformation is equivalent to transforming the original signal and the correction separately and then combining them. The matrix-sign map, however, is nonlinear, so these two constructions generally produce different update directions. This distinction motivates considering update-level refinement in addition to modifying the momentum before orthogonalization.

The ADMM-Inspired Momentum (AIM) framework (Ren et al., 2026) provides a systematic view of optimizer design by separating momentum construction, update geometry, and parameter-subproblem approximation. Momentum first defines a tentative step, which can then be refined through the parameter subproblem. Under Euclidean geometry and suitable parameter choices, direct assignment of the tentative point recovers momentum descent, whereas a one-step fixed-point refinement yields a

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.

Nesterov-style update. Under spectral geometry, direct assignment instead gives Muon's matrix-sign direction with a nuclear-norm weight. These correspondences suggest that refining the parameter subproblem provides an independent route to acceleration: the momentum state continues to summarize gradient history, while the discrepancy between the current gradient and momentum can be used to refine the final parameter update.

Motivated by this perspective, we propose **AIMuon** (**A**DMM-**I**nspired **M**oment**u**m **O**rthogonalized by **N**ewton–Schulz), obtained by extending AIM's parameter-subproblem refinement to the spectral geometry underlying Muon. AIMuon retains Muon's momentum-based direction and adds a correction driven by the gradient–momentum residual, which is transformed separately by matrix sign. The momentum and residual directions are combined using matrix-wise nuclear-norm weights that adapt their relative strengths across parameter matrices, and the combined direction is rescaled to the corresponding Muon update magnitude using its RMS-matching convention (Liu et al., 2025). This yields a principled refinement of the Muon direction using the same spectral map for the momentum and residual terms.

Our contributions are summarized as follows:

- We propose **AIMuon**, a residual-corrected matrix optimizer derived by extending AIM's parameter-subproblem refinement to Muon's spectral geometry. AIMuon introduces a separately transformed gradient–momentum residual correction and matrix-wise nuclear-norm weighting, refining the standard Muon direction while preserving its momentum recursion.
- We establish a stochastic nonconvex stationarity bound for the exact residual-corrected iteration underlying AIMuon, characterizing both the optimization error and the variance-dependent stochastic error.
- On GPT-2-small and GPT-2-medium, AIMuon achieves lower final validation loss than Muon and MONA and reaches Muon's final validation loss with approximately 12.8% and 7.7% fewer optimization steps, respectively. The improvement also persists in wall-clock time on GPT-2-medium. Ablation and sensitivity studies further characterize the contributions of the residual correction, matrix-wise nuclear-norm weighting, final normalization, and the correction coefficient.

2 Background and Related Work

In this section, we first review optimizer designs related to momentum construction and update refinement, and then introduce the spectral properties underlying AIMuon. Details of the general AIM framework are provided in Appendix D.

2.1 Related Work

Momentum and adaptivity. Momentum and adaptivity are two important components in the design of modern optimization algorithms. Classical Polyak momentum (Polyak, 1964) uses an exponential average of past gradients, while Nesterov acceleration (Nesterov, 1983) modifies how momentum information contributes to the update. Adaptive methods further rescale updates using gradient statistics: Adam (Kingma & Ba, 2015) combines first- and second-moment estimates, NAdam (Dozat, 2016) incorporates Nesterov momentum into this adaptive framework, and AdamW (Loshchilov & Hutter, 2019) decouples weight decay from adaptive scaling. More recent methods such as RAD (Lyu et al., 2025) derive adaptive updates from alternative dynamical principles. Building on RAD's adaptive preconditioner within the AIM framework, RADAR (Ren et al., 2026) further modifies both the momentum construction and the parameter-update subproblem. These developments highlight the central role of momentum construction and adaptive scaling in shaping the update.

Structured preconditioning and normalization. Beyond coordinate-wise adaptivity, many optimizers exploit structure within matrix or tensor parameters. Adafactor (Shazeer & Stern, 2018) approximates matrix-valued second-moment estimates using row- and column-wise factors, reducing optimizer memory, Adam-mini (Zhang et al., 2025) replaces Adam's element-wise adaptive learning rates with shared learning rates over carefully chosen parameter blocks, and Shampoo (Gupta

et al., 2018) builds Kronecker-factored second-order preconditioners along the dimensions of tensor parameters. SOAP (Vyas et al., 2025) applies Adam-style adaptivity in a rotated basis induced by Shampoo-style preconditioners. Matrix-geometric methods such as spectral descent (Carlson et al., 2015), Muon, and PolarGrad (Lau et al., 2025) exploit spectral or polar structure to construct matrix-aware update directions, with PolarGrad further introducing nuclear-norm-based scaling. Building on this line, NorMuon (Li et al., 2026b) augments Muon with neuron-wise adaptive normalization based on second-moment statistics, while AdaMuon (Si et al., 2025) combines element-wise adaptivity with orthogonalized matrix updates. This line of work broadens optimizer design from coordinate-wise scaling to structured matrix- and tensor-level transformations.

Scalable orthogonalized optimizers. Recent work has focused on improving the efficiency and scalability of orthogonalized matrix updates. Liu et al. (2025) study the scaling, update normalization, and distributed implementation of Muon for large-scale language-model training, Dion (Ahn et al., 2025) reduces computation and communication through low-rank orthonormalized updates with error feedback, and ROOT (He et al., 2025) adapts its orthogonalization coefficients to matrix dimensions and uses a proximal formulation to improve robustness to outlier noise. These efforts extend orthogonalized optimization toward more practical large-scale training settings.

2.2 Matrix-Sign Geometry

Consider the stochastic optimization problem

$$\min_{\Theta} \; F(\Theta) := \mathbb{E}_{\xi}[f(\Theta;\xi)]\,, \tag{1}$$

with stochastic gradient $G_k = \nabla f(\Theta_k;\xi_k)$. For compatible matrices, we use $\langle A, B\rangle = \mathrm{tr}(A^\top B)$ for the Frobenius inner product and denote the Frobenius, spectral, and nuclear norms by $\|\cdot\|_F$, $\|\cdot\|_2$, and $\|\cdot\|_*$, respectively.

Let $A = U_r\Sigma_r V_r^\top$ be the compact SVD of A. For nonzero A, define the matrix-sign, or polar, direction as

$$\mathrm{MSign}(A) := U_r V_r^\top, \tag{2}$$

and set $\mathrm{MSign}(0) = 0$. This direction is naturally associated with spectral–nuclear duality:

$$\|A\|_* = \max_{\|Z\|_2\le 1} \langle A, Z\rangle, \qquad \langle A, \mathrm{MSign}(A)\rangle = \|A\|_*. \tag{3}$$

For the spectral-norm penalized problem

$$\min_{Z} \left\{ \langle A, Z\rangle + \frac{\rho}{2}\|Z\|_2^2 \right\}, \qquad \rho > 0, \tag{4}$$

the full solution set is characterized in Appendix A. Choosing the canonical polar representative gives

$$Z^\star_{\mathrm{ms}} = -\frac{\|A\|_*}{\rho}\,\mathrm{MSign}(A). \tag{5}$$

Thus, the matrix-sign factor determines the spectral direction, while the nuclear norm determines its radial magnitude. AIMuon applies this spectral construction to both the momentum and the residual correction.

2.3 Direct Assignment and the Muon Direction

Within AIM, choosing the squared spectral-norm penalty $\psi(X) = \|X\|_2^2$ makes the auxiliary-variable update take the form

$$\min_{\Delta} \left\{ \langle M_k, \Delta\rangle + \frac{\rho_{1,k}}{2}\|\Delta\|_2^2 \right\}, \qquad \Delta = Y - \Theta_k. \tag{6}$$

Using equation 5, the canonical spectral step is

$$Y_{k+1} = \Theta_k - \frac{\|M_k\|_*}{\rho_{1,k}}\,\mathrm{MSign}(M_k). \tag{7}$$

Direct assignment $\Theta_{k+1} = Y_{k+1}$ with $\rho_{1,k} = 1/\eta_k$ then yields

$$\Theta_{k+1} = \Theta_k - \eta_k \|M_k\|_* \operatorname{MSign}(M_k). \tag{8}$$

This has the same polar-direction and nuclear-norm scaling structure as PolarGrad (Lau et al., 2025), with momentum used as the input matrix. Muon retains the same matrix-sign direction but replaces the exact nuclear-norm radial factor with its practical matrix-wise normalization; under the RMS-matching convention used to align Muon's update scale with AdamW (Liu et al., 2025), this gives the practical Muon update.

Direct assignment stops at this spectral momentum step. AIMuon instead retains the parameter-subproblem refinement and carries it into the same matrix-sign geometry, leading to the residual correction developed in Sec. 3.

3 AIMuon: Spectral Residual Correction

AIMuon extends Muon by complementing its momentum-based matrix-sign update with an independently orthogonalized correction derived from the gradient–momentum residual. We first introduce the resulting update and its practical implementation, and then show how the residual correction and its spectral weighting naturally arise from the AIM parameter subproblem.

3.1 The Residual-Corrected Update

For the current stochastic gradient G_k, we retain the standard EMA momentum recursion

$$M_k = \beta M_{k-1} + (1-\beta) G_k, \qquad M_{-1} = 0. \tag{9}$$

We define the gradient–momentum residual as $E_k := G_k - M_k$. In the original parameter space, adding a scaled residual directly to the momentum gives

$$M_k + cE_k = (1-c)M_k + cG_k. \tag{10}$$

Thus, before the nonlinear matrix-sign transformation, the residual simply adjusts the relative contribution of the momentum and current gradient; in particular, $c = 1-\beta$ yields the corresponding Nesterov-style combination. AIMuon instead treats E_k as a separate correction signal and transforms the momentum and residual independently. The coefficient c therefore controls the strength of this residual correction after the two components are mapped into matrix-sign geometry.

The exact AIMuon update applies the same nuclear-weighted matrix-sign map to the momentum and residual:

$$\Theta_{k+1} = \Theta_k - \eta_k \left[\|M_k\|_* \operatorname{MSign}(M_k) + c \, \|E_k\|_* \operatorname{MSign}(E_k) \right]. \tag{11}$$

Here η_k denotes the learning rate. The two components are transformed separately, with each matrix-sign direction scaled by the nuclear norm of its corresponding matrix.

Algorithm 1 summarizes the practical update for a parameter matrix $\Theta_k \in \mathbb{R}^{m\times n}$. Here $\varepsilon > 0$ stabilizes the ratios and λ_{wd} denotes weight decay.

Practical implementation. In practice, we approximate the two matrix-sign directions independently using five Newton–Schulz iterations:

$$D_k^M = \mathrm{NS}_5(M_k), \qquad D_k^E = \mathrm{NS}_5(E_k). \tag{12}$$

The resulting residual direction complements the standard Muon update, with both terms constructed using the same spectral map. Sec. 3.2 derives both the residual signal and its spectral weighting from the AIM parameter subproblem.

Matrix-wise nuclear-norm weighting. Direct computation of the nuclear norm requires a singular-value decomposition and is expensive for large-scale training. Since the matrix-sign directions are already approximated by the Newton–Schulz iterations, we estimate the corresponding nuclear norms as

$$\widehat{\nu}_k^M = \left| \langle M_k, D_k^M \rangle \right|, \qquad \widehat{\nu}_k^E = \left| \langle E_k, D_k^E \rangle \right|. \tag{13}$$

Algorithm 1 AIMuon

Require: Θ_0, $M_{-1} = 0$, learning rates $\{\eta_k\}_{k=0}^{T-1}$, momentum coefficient β, correction coefficient c, weight decay λ_{wd}

for $k = 0, \ldots, T-1$ **do**
  $G_k \leftarrow \nabla f(\Theta_k; \xi_k)$
  $M_k \leftarrow \beta M_{k-1} + (1-\beta) G_k$
  $E_k \leftarrow G_k - M_k$
  $D_k^M \leftarrow \mathrm{NS}_5(M_k)$
  $D_k^E \leftarrow \mathrm{NS}_5(E_k)$
  $\widehat{\nu}_k^M \leftarrow |\langle M_k, D_k^M \rangle|$
  $\widehat{\nu}_k^E \leftarrow |\langle E_k, D_k^E \rangle|$
  $\lambda_k \leftarrow c \frac{\widehat{\nu}_k^E}{\widehat{\nu}_k^M + \varepsilon}$
  $\widetilde{O}_k \leftarrow \widehat{\nu}_k^M \left(D_k^M + \lambda_k D_k^E \right)$
  $O_k \leftarrow 0.2\sqrt{\max(m,n)} \frac{\|D_k^M\|_F}{\|\widetilde{O}_k\|_F + \varepsilon} \widetilde{O}_k$
  $\Theta_{k+1} \leftarrow (1 - \eta_k \lambda_{wd}) \Theta_k - \eta_k O_k$
end for

This reuses the computed NS5 directions and avoids the cost of an additional SVD. For each parameter matrix, we then define the relative strength of the residual correction as

$$\lambda_k = c \frac{\widehat{\nu}_k^E}{\widehat{\nu}_k^M + \varepsilon}. \tag{14}$$

The two directions are then combined as $\widetilde{O}_k = \widehat{\nu}_k^M \left(D_k^M + \lambda_k D_k^E \right)$. Since λ_k depends on the residual-to-momentum nuclear-norm ratio for each parameter matrix, the relative contribution of the residual direction adapts across matrices and throughout training.

RMS matching. Although D_k^M and D_k^E are individually approximately orthogonalized, their weighted combination $\widetilde{O}_k$ is generally no longer a matrix-sign direction. Its Frobenius norm can therefore vary with the nuclear-norm weights and the alignment between the two components. Following Muon's RMS-matching convention (Liu et al., 2025), we rescale the combined direction as

$$O_k = 0.2\sqrt{\max(m,n)} \frac{\|D_k^M\|_F}{\|\widetilde{O}_k\|_F + \varepsilon} \widetilde{O}_k. \tag{15}$$

This approximately restores the overall update magnitude to the standard Muon scale. Moreover, up to the stabilizing ε term, the common momentum nuclear-norm factor $\widehat{\nu}_k^M$ cancels under this normalization. Consequently, the nuclear-norm information enters the practical update primarily through the matrix-wise ratio $\widehat{\nu}_k^E / (\widehat{\nu}_k^M + \varepsilon)$ in λ_k, which adaptively controls the relative contribution of the residual direction.

3.2 Residual Correction from the AIM Parameter Subproblem

The spectral auxiliary-variable subproblem admits the matrix-sign representative

$$Y_{k+1} = \Theta_k - \frac{\|M_k\|_*}{\rho_{1,k}} \mathrm{MSign}(M_k). \tag{16}$$

As discussed in Sec. 2.3, directly setting $\Theta_{k+1} = Y_{k+1}$ yields the Muon-type update.

AIMuon instead retains the subsequent parameter subproblem,

$$\min_{\Theta} \left\{ F(\Theta) + \langle M_k, Y_{k+1} - \Theta \rangle + \frac{\rho_{2,k}}{2} \|Y_{k+1} - \Theta\|_2^2 \right\}. \tag{17}$$

Using Θ_k to form a one-step local approximation and linearizing F with the same stochastic gradient G_k, the parameter subproblem becomes

$$\min_{\Theta} \left\{ \langle E_k, \Theta - Y_{k+1} \rangle + \frac{\rho_{2,k}}{2} \|\Theta - Y_{k+1}\|_2^2 \right\}, \tag{18}$$

where $E_k := G_k - M_k$ is the gradient–momentum residual.

Applying the same spectral geometry to this residual-driven subproblem gives

$$\Theta_{k+1} = Y_{k+1} - \frac{\|E_k\|_*}{\rho_{2,k}} \operatorname{MSign}(E_k). \tag{19}$$

Substituting equation 16 yields

$$\Theta_{k+1} = \Theta_k - \frac{1}{\rho_{1,k}} \left[\|M_k\|_* \operatorname{MSign}(M_k) + \frac{\rho_{1,k}}{\rho_{2,k}} \|E_k\|_* \operatorname{MSign}(E_k) \right]. \tag{20}$$

Let $\eta_k = 1/\rho_{1,k}$ and $c = \rho_{1,k}/\rho_{2,k}$. Then equation 20 recovers the AIMuon update in equation 11. This shows that the AIM parameter refinement naturally produces a separately transformed residual correction, with c determined by the relative strength of the two subproblem penalties.

3.3 Stochastic Convergence of the Exact Update

We establish a stochastic Frobenius-stationarity bound for the exact residual-corrected iteration in equation 11. For constant step size and batch size, the bound consists of an $\mathcal{O}(T^{-1})$ optimization term and an $\mathcal{O}(B^{-1})$ stochastic error floor. The assumptions, precise statement, and proof are given in Appendix B.

4 Experiments

We evaluate AIMuon in terms of optimization performance, wall-clock efficiency, component contributions, and residual-correction dynamics, covering both its effectiveness and practical cost.

4.1 Setup

Models and datasets. We evaluate AIMuon on two language-model pretraining settings: GPT-2-small and GPT-2-medium (Radford et al., 2019). GPT-2-small is pretrained from scratch on FineWeb-Edu, while GPT-2-medium is pretrained from scratch on FineWeb (Penedo et al., 2024). The two models are trained for approximately 2.48B and 3.55B tokens, respectively.

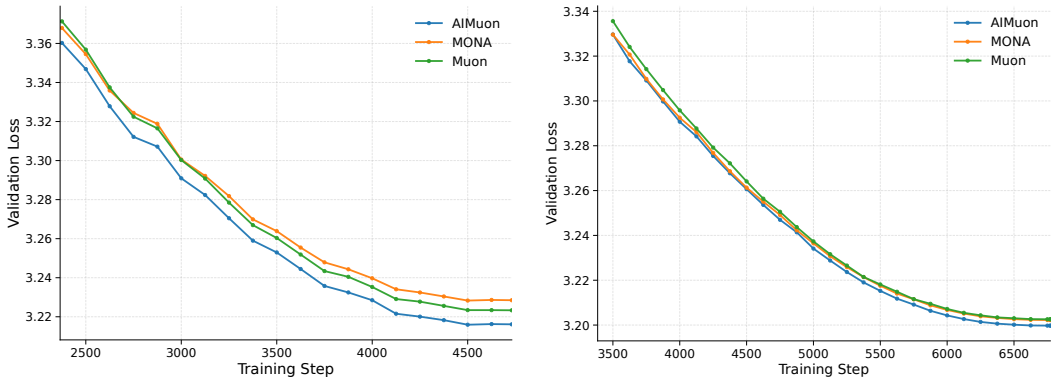
Compared methods. We compare AIMuon with Muon and MONA. Muon serves as the direct baseline for evaluating the effect of the proposed residual correction, while MONA provides an additional comparison with a Muon variant that modifies the momentum construction. Within each setting, methods share the initialization and data order, with learning rates selected independently. Detailed training configurations, optimizer settings, learning-rate selection, and evaluation protocol are provided in Appendix C.

4.2 Optimization Performance

Figure 1 compares the validation trajectories of AIMuon, Muon, and MONA on GPT-2-small and GPT-2-medium. Across both settings, AIMuon achieves a lower final validation loss than the two baselines and reaches Muon's final loss with fewer optimization steps.

GPT-2-small. AIMuon achieves a final validation loss of 3.2161, compared with 3.2233 for Muon and 3.2284 for MONA. It reaches Muon's final validation loss at step 4125, compared with Muon's full training budget of 4731 steps, corresponding to a 12.8% reduction in optimization steps.

GPT-2-medium. AIMuon achieves a final validation loss of 3.1997, compared with 3.2026 for Muon and 3.2022 for MONA. It reaches Muon's final validation loss at step 6250 of the 6772-step training budget, corresponding to a 7.7% reduction in optimization steps. The improvement is also observed in the larger GPT-2-medium setting, which uses a different dataset and training budget.



(a) GPT-2-small.

(b) GPT-2-medium.

Figure 1: Validation loss over the final 50% of pretraining on GPT-2-small with FineWeb-Edu and GPT-2-medium with FineWeb. Validation loss is shown without smoothing.

4.3 Wall-Clock Efficiency

AIMuon performs an additional NS5 computation for the residual direction, increasing the per-step computational cost relative to Muon. To account for this overhead, we compare validation loss against elapsed training time for GPT-2-medium on four RTX 5090 GPUs.

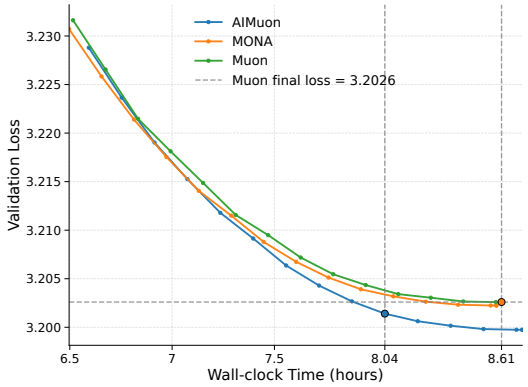


Figure 2: Validation loss versus elapsed training time for GPT-2-medium on FineWeb over the final 25% of training, including AIMuon's additional residual NS5 computation. The horizontal dashed line marks Muon's final validation loss. The vertical dashed lines mark AIMuon's first evaluated crossing of this target and the completion of Muon's full run.

As shown in Figure 2, AIMuon first reaches Muon's final validation loss of 3.2026 at approximately 8.04 hours, compared with 8.61 hours for Muon's full run. This corresponds to approximately 6.6% less elapsed time, showing that the reduction in optimization steps translates into a wall-clock benefit despite the additional residual NS5 computation. The time reduction is smaller than the 7.7% reduction in optimization steps, consistent with the additional residual computation. These measurements use the original full-budget learning-rate schedules.

4.4 Component Analysis

We examine the main components of AIMuon on GPT-2-small: the residual correction itself, the separate transformation of the residual, the matrix-wise adaptation of its relative weight, and the final RMS matching. Figure 3 and Table 1 summarize the resulting variants. All experiments follow the main training setup; detailed update equations and coefficients are provided in Appendix C.

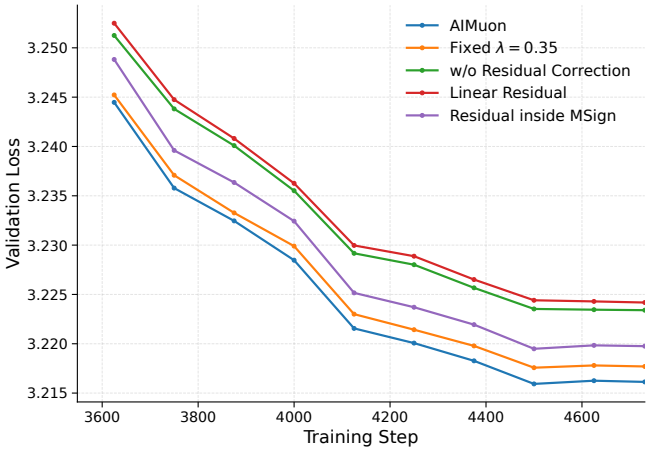


Figure 3: Component comparisons over the final 25% of GPT-2-small training. The fixed-weight variant uses $\lambda = 0.35$. The no-F-norm-ratio variant is reported only in Table 1; all other variants retain the final RMS matching.

Table 1: Final validation loss on GPT-2-small.

Configuration	Loss
AIMuon	3.2161
Fixed $\lambda = 0.35$	3.2177
No residual	3.2234
Linear residual	3.2242
Inside MSign	3.2198
No F-norm ratio	3.5072

Residual correction. We first remove the residual branch by setting $c = 0$, while leaving the remaining update unchanged. The final validation loss increases from 3.2161 to 3.2234. This comparison isolates the contribution of the gradient–momentum residual beyond the standard momentum matrix-sign direction.

Separate residual transformation. We next compare the separately transformed residual with two alternative ways of incorporating residual information. The *Inside MSign* variant applies the matrix-sign transformation to $M_k + 0.05E_k$, whereas the *Linear residual* variant adds the raw term $0.05E_k$ to the transformed momentum direction. Both variants retain the final RMS matching. Their final validation losses are 3.2198 and 3.2242, respectively, compared with 3.2161 for AIMuon. These results indicate that, under the evaluated configurations, applying a separate matrix-sign transformation to the residual is beneficial relative to these two alternatives.

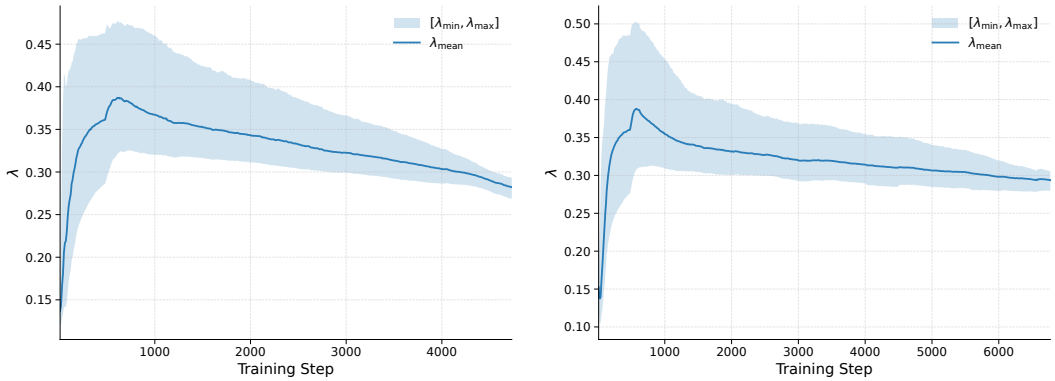
Matrix-wise residual weighting. We then examine the effect of the adaptive nuclear-norm ratio by replacing λ_k with the fixed value 0.35, chosen to approximately match the mean coefficient observed around the middle of training. The momentum and residual remain separately transformed and the same final RMS matching is retained. The fixed-weight variant obtains a final validation loss of 3.2177, compared with 3.2161 for AIMuon. AIMuon thus achieves a lower loss than this representative fixed-weight variant.

Final RMS matching. Finally, we remove the Frobenius-norm ratio from the final rescaling while retaining the dimension factor $0.2\sqrt{\max(m, n)}$. The final validation loss increases substantially to 3.5072. This result shows that removing the final normalization substantially degrades performance under the learning rate and training configuration used here.

4.5 Residual Correction Dynamics and Sensitivity

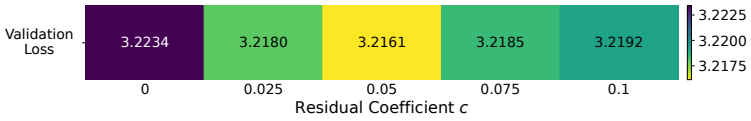
We next examine how the residual correction evolves during training and how performance depends on the base coefficient c.

Adaptation during training. Figures 4a–4b track the effective matrix-wise coefficient λ_k throughout training. For both GPT-2-small and GPT-2-medium, the mean coefficient rises early in training and then gradually decreases, while the cross-matrix range narrows toward the end of training. Since $\lambda_k = c\,\widehat{\nu}_k^E/(\widehat{\nu}_k^M + \varepsilon)$, these dynamics reflect changes in the estimated residual-to-momentum nuclear-norm ratio across both parameter matrices and training stages. Thus, the contribution of the residual correction is not fixed throughout training but adapts to the evolving optimization dynamics.



(a) Effective coefficient, GPT-2-small.

(b) Effective coefficient, GPT-2-medium.



(c) Sensitivity to c, GPT-2-small.

Figure 4: Residual-correction dynamics and sensitivity. (a–b) Mean and cross-matrix range of the effective coefficient λ_k, shown with a 50-point moving average. (c) Final validation loss under different values of c, with all other hyperparameters fixed.

The base coefficient c sets a shared reference scale and remains fixed throughout each run; matrix-wise differences and temporal changes in λ_k come from the estimated norm ratio. The sensitivity analysis below evaluates the effect of changing this reference scale across separate runs while retaining the same rule for adapting the matrix-wise weights.

Sensitivity to c. We further vary c on GPT-2-small while keeping all other hyperparameters fixed. As shown in Figure 4c, each tested nonzero value $c \in \{0.025, 0.05, 0.075, 0.1\}$ yields a lower final validation loss than the residual-free setting $c = 0$. The preselected value $c = 0.05$ achieves the lowest loss of 3.2161, compared with 3.2234 at $c = 0$. Performance remains improved over the residual-free baseline across the tested range, suggesting that the benefit of the residual correction is not confined to a single choice of c.

5 Conclusion

We introduced AIMuon, a residual-corrected matrix optimizer that refines Muon's spectral update while retaining its momentum recursion. AIMuon applies separate matrix-sign transformations to the momentum and gradient–momentum residual, combines the resulting directions using matrix-wise nuclear-norm weighting, and rescales the combined update to match Muon's magnitude. We established a stochastic stationarity bound for the corresponding exact iteration. On GPT-2-small and GPT-2-medium pretraining, AIMuon achieves lower final validation loss than Muon and MONA and reaches Muon's final validation loss with 12.8% and 7.7% fewer optimization steps, respectively. Together with the component and sensitivity analyses, these results show that refining the parameter update provides a complementary direction for improving matrix-sign optimization beyond modifying momentum construction.

Limitations and outlook. Our empirical study focuses on Muon-style language-model pretraining, providing a controlled setting for examining spectral parameter refinement; its behavior across other model families and optimization regimes remains to be explored. Our analysis characterizes the exact residual-corrected spectral iteration, while extending the theory to finite Newton–Schulz approximation and RMS matching would further connect the analysis with practical implementations.

AI USE STATEMENT

Generative AI tools were used solely as auxiliary tools for language polishing and assistance with experimental and plotting code. The authors reviewed the resulting text and code and take full responsibility for 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. URL https://arxiv.org/abs/2504.05295.

David E. Carlson, Edo Collins, Ya-Ping Hsieh, Lawrence Carin, and Volkan Cevher. Preconditioned spectral descent for deep learning. In *Advances in Neural Information Processing Systems*, volume 28, pp. 2971–2979, 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/hash/f50a6c02a3fc5a3a5d4d9391f05f3efc-Abstract.html.

Timothy Dozat. Incorporating Nesterov momentum into Adam. In *International Conference on Learning Representations Workshop Track*, 2016.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1842–1850, 2018. URL https://proceedings.mlr.press/v80/gupta18a.html.

Wei He, Kai Han, Hang Zhou, Hanting Chen, Zhicheng Liu, Xinghao Chen, and Yunhe Wang. ROOT: Robust orthogonalized optimizer for neural network training. *arXiv preprint arXiv:2511.20626*, 2025. URL https://arxiv.org/abs/2511.20626.

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

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

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

Jiacheng Li, Jianchao Tan, Hongtao Xu, Jiaqi Zhang, Yifan Lu, Yerui Sun, Yuchen Xie, and Xunliang Cai. MONA: Muon optimizer with Nesterov acceleration for scalable language model training. *arXiv preprint arXiv:2605.26842*, 2026a.

Zichong Li, Liming Liu, Chen Liang, Weizhu Chen, and Tuo Zhao. NorMuon: Making Muon more efficient and scalable. In *International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=m1IRWFAMsa.

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.

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

Yao Lyu, Xiangteng Zhang, Shengbo Eben Li, Jingliang Duan, Letian Tao, Qing Xu, Lei He, and Keqiang Li. Conformal symplectic optimization for stable reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):11049–11063, 2025. doi: 10.1109/TNNLS.2024.3511670.

Yurii E Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Doklady Akademii Nauk SSSR*, 269(3):543–547, 1983.

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. URL https://proceedings.nips.cc/paper_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets_and_Benchmarks_Track.html.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Zihan Qiu, Zekun Wang, Xiao Li, Yanpeng Li, Yang Xu, Yixuan Wang, Huaqing Zhang, Rui Men, Bochao Mao, Chengruidong Zhang, Fan Zhou, Hao Luo, Haofeng Huang, Haoran Lian, Haoyan Huang, Hongqing Chen, Jianwei Zhang, Jing Xu, Junjie Wang, Langshi Chen, Liangyu Wang, Linlang Jiang, Man Yuan, Minmin Sun, Peng Jin, Siqi Zhang, Siyu Wang, Xingzhang Ren, Yakai Wang, Yi Zhang, Yiming Dong, Yizhong Cao, Yubo Ma, Yunfei Mao, Bo Zheng, and Dayiheng Liu. On the design of Qwen3.8-Next architecture: Evaluation, efficiency, and training stability. *arXiv preprint arXiv:2608.30320*, 2026.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Zhixin Ren, Yao Lyu, Congrong Li, Liping Zhang, and Shengbo Eben Li. Momentum as residual-driven multiplier correction for deep learning optimization. *arXiv preprint arXiv:2608.12925*, 2026.

Günther Schulz. Iterative berechung der reziproken matrix. *Zeitschrift für Angewandte Mathematik und Mechanik*, 13(1):57–59, 1933. doi: 10.1002/zamm.19330130111.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4596–4604, 2018. URL https://proceedings.mlr.press/v80/shazeer18a.html.

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

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. SOAP: Improving and stabilizing Shampoo using Adam for language modeling. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=IDxZhXrpNf.

Xingyu Xie, Pan Zhou, Huan Li, Zhouchen Lin, and Shuicheng Yan. Adan: Adaptive Nesterov momentum algorithm for faster optimizing deep models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9508–9520, 2024. doi: 10.1109/TPAMI.2024.3423382.

Yushun Zhang, Congliang Chen, Ziniu Li, Tian Ding, Chenwei Wu, Diederik P. Kingma, Yinyu Ye, Zhi-Quan Luo, and Ruoyu Sun. Adam-mini: Use fewer learning rates to gain more. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=iBExhaU3Lc.

Tong Zhao, Jiacheng Li, Yuanchang Zhou, Guangming Tan, and Weile Jia. Exploring landscapes for better minima along valleys. In *Advances in Neural Information Processing Systems*, volume 38, pp. 171496–171547, 2025. doi: 10.52202/085713-5709. URL https://proceedings.nips.cc/paper_files/paper/2025/hash/fa84d72c79e28d4bf58f6c7ac92cd51c-Abstract-Conference.html.

A SOLUTION SET OF THE SPECTRAL-NORM SUBPROBLEM

We characterize the complete solution set of the spectral-norm penalized subproblem in equation 4. The matrix-sign update used in the main text is then obtained by selecting a canonical representative from this set.

Proposition 1 (Solution set of the spectral-norm subproblem)**.** *Let $A \in \mathbb{R}^{m\times n}$ and $\rho > 0$, and consider*

$$\min_Z \left\{ \langle A, Z\rangle + \frac{\rho}{2}\|Z\|_2^2 \right\}. \tag{21}$$

If $A = 0$, the unique minimizer is $Z^\star = 0$. Otherwise, let $r = \operatorname{rank}(A)$ and write the compact SVD as $A = U_r\Sigma_r V_r^\top$. Let $U_\perp$ and $V_\perp$ denote orthonormal bases for the corresponding orthogonal complements. Then the optimal value is

$$-\frac{\|A\|_*^2}{2\rho}, \tag{22}$$

and the complete set of minimizers is

$$\begin{aligned}&\arg\min_Z \left\{ \langle A, Z\rangle + \frac{\rho}{2}\|Z\|_2^2 \right\} \\ &\quad = \left\{ \frac{\|A\|_*}{\rho}\left(-U_rV_r^\top + U_\perp Q V_\perp^\top\right) \;\middle|\; Q \in \mathbb{R}^{(m-r)\times(n-r)},\ \|Q\|_2 \le 1 \right\}.\end{aligned} \tag{23}$$

Proof. The case $A = 0$ is immediate, since the objective reduces to $\rho\|Z\|_2^2/2$. We therefore assume $A \neq 0$. Let

$$A = U_r\Sigma_r V_r^\top, \qquad \Sigma_r = \operatorname{diag}(\sigma_1, \dots, \sigma_r),$$

with $\sigma_i > 0$, and set $a := \|A\|_* > 0$.

For any Z, let $t = \|Z\|_2$. By spectral–nuclear duality,

$$\langle A, Z\rangle \ge -\|A\|_*\|Z\|_2 = -at. \tag{24}$$

Hence

$$\begin{aligned}\langle A, Z\rangle + \frac{\rho}{2}\|Z\|_2^2 &\ge -at + \frac{\rho}{2}t^2 \\ &= \frac{\rho}{2}\left(t - \frac{a}{\rho}\right)^2 - \frac{a^2}{2\rho} \\ &\ge -\frac{a^2}{2\rho}.\end{aligned} \tag{25}$$

The lower bound is attained by $Z = -(a/\rho)U_rV_r^\top$, since $\|U_rV_r^\top\|_2 = 1$ and $\langle A, U_rV_r^\top\rangle = a$. This proves equation 22.

It remains to characterize all points attaining this bound. Equality in equation 25 requires

$$\|Z\|_2 = \frac{a}{\rho}, \qquad \langle A, Z\rangle = -\frac{a^2}{\rho}. \tag{26}$$

Define

$$C := \frac{\rho}{a}Z.$$

Then every minimizer satisfies

$$\|C\|_2 = 1, \qquad \langle A, C\rangle = -a. \tag{27}$$

Let u_i and v_i denote the singular vectors associated with σ_i. From equation 27,

$$\sum_{i=1}^r \sigma_i u_i^\top C v_i = -\sum_{i=1}^r \sigma_i.$$

Since $\|C\|_2 = 1$, each $u_i^\top C v_i \ge -1$. Because every σ_i is positive, equality of the weighted sums forces

$$u_i^\top C v_i = -1, \qquad i = 1, \dots, r.$$

Moreover, $\|Cv_i\|_2 \le 1$, so

$$\|Cv_i + u_i\|_2^2 = \|Cv_i\|_2^2 + 2u_i^\top Cv_i + 1 \le 0.$$

Therefore $Cv_i = -u_i$ for every i. Applying the same argument to $C^\top$ gives $C^\top u_i = -v_i$. Equivalently,

$$CV_r = -U_r, \qquad C^\top U_r = -V_r. \tag{28}$$

Extend U_r and V_r to orthogonal matrices $U = [U_r \ U_\perp]$ and $V = [V_r \ V_\perp]$. The relations in equation 28 imply that C has the block form

$$U^\top CV = \begin{pmatrix} -I_r & 0 \\ 0 & Q \end{pmatrix}, \qquad Q = U_\perp^\top CV_\perp. \tag{29}$$

Since $\|C\|_2 = 1$, necessarily $\|Q\|_2 \le 1$. Transforming back gives

$$C = -U_r V_r^\top + U_\perp Q V_\perp^\top, \tag{30}$$

and hence

$$Z = \frac{\|A\|_*}{\rho}\left(-U_r V_r^\top + U_\perp Q V_\perp^\top\right), \qquad \|Q\|_2 \le 1. \tag{31}$$

Conversely, any matrix of this form is optimal. Indeed, for any Q with $\|Q\|_2 \le 1$, the block representation equation 29 gives

$$\|C\|_2 = \max\{1, \|Q\|_2\} = 1,$$

while the orthogonality of the complementary subspaces gives $\langle A, C\rangle = -a$. Thus equation 27 holds, so equation 26 holds as well. Therefore every such Z attains the optimal value in equation 22, completing the characterization in equation 23. □

Choosing $Q = 0$ in equation 23 gives

$$Z_{\text{can}} = -\frac{\|A\|_*}{\rho} U_r V_r^\top = -\frac{\|A\|_*}{\rho}\,\mathrm{MSign}(A), \tag{32}$$

which is the canonical representative used in equation 5. If $r = \min(m, n)$, one of the complementary subspaces is trivial, so the additional term vanishes and the minimizer is unique. If $0 < r < \min(m, n)$, different admissible choices of Q yield distinct minimizers.

B Stochastic Convergence Analysis

We analyze the exact AIMuon iteration, excluding Newton–Schulz approximation errors, Frobenius-norm matching, gradient clipping, weight decay, and auxiliary AdamW updates. For a matrix $A \in \mathbb{R}^{m\times n}$, define

$$\mathcal{D}(A) := \|A\|_*\,\mathrm{MSign}(A), \qquad \mathcal{D}(0) := 0, \tag{33}$$

where $\mathrm{MSign}(A) = U_q V_q^\top$ for the compact SVD of A containing its $q = \mathrm{rank}(A)$ positive singular values. We consider the exact iteration

$$\begin{aligned} M_k &= \beta M_{k-1} + (1-\beta)G_k, \\ E_k &= G_k - M_k, \\ \mathcal{U}_k &= \mathcal{D}(M_k) + c\mathcal{D}(E_k), \\ \Theta_{k+1} &= \Theta_k - \eta_k \mathcal{U}_k, \qquad M_{-1} = 0. \end{aligned} \tag{34}$$

We take Θ_0 to be deterministic. For the single-matrix iteration considered here, a fixed positive scaling factor can be absorbed into η_k.

B.1 Assumptions and Main Result

Assumption 1 (Lower boundedness and smoothness)**.** *The objective $F : \mathbb{R}^{m\times n} \to \mathbb{R}$ is continuously differentiable, bounded below by $F_{\inf} > -\infty$, and has an L_F-Lipschitz continuous gradient for some $L_F > 0$:*

$$\|\nabla F(X) - \nabla F(Y)\|_F \le L_F \|X - Y\|_F, \qquad \forall X, Y. \tag{35}$$

Assumption 2 (Unbiased stochastic gradients)**.** *Let $\mathcal{F}_k$ denote the filtration generated by the history prior to sampling G_k, so that Θ_k and M_{k-1} are $\mathcal{F}_k$-measurable. For deterministic batch sizes $B_k \ge 1$, assume*

$$\begin{aligned} \mathbb{E}[G_k \mid \mathcal{F}_k] &= \nabla F(\Theta_k), \\ \mathbb{E}\big[\|G_k - \nabla F(\Theta_k)\|_F^2 \,\big|\, \mathcal{F}_k\big] &\le \frac{\sigma_F^2}{B_k}. \end{aligned} \tag{36}$$

Theorem 1 (Stochastic Frobenius-stationarity bound)**.** *Suppose Assumptions 1 and 2 hold. Let $0 \le \beta < 1$ and $0 < c < 4$. Define*

$$r := \min(m, n), \qquad \kappa_c := \frac{4 - c}{4(1 + 2c)}, \qquad \Delta_F := F(\Theta_0) - F_{\inf}. \tag{37}$$

Assume that the stepsizes $\{\eta_k\}$ are deterministic, nonnegative, and nonincreasing, with

$$0 < \eta_0 \le \frac{\kappa_c}{r L_F}, \qquad S_T := \sum_{k=0}^{T-1} \eta_k > 0. \tag{38}$$

Then the iterates generated by equation 34 satisfy

$$\begin{aligned} \frac{1}{S_T} \sum_{k=0}^{T-1} \eta_k \mathbb{E}\|\nabla F(\Theta_k)\|_F^2 \le \frac{4\Delta_F}{c\kappa_c S_T} \\ + \frac{4r\sigma_F^2}{c\kappa_c^2 S_T} \sum_{k=0}^{T-1} \frac{\eta_k}{B_k}. \end{aligned} \tag{39}$$

In particular, for constant stepsize $\eta_k \equiv \eta > 0$ and constant batch size $B_k \equiv B$,

$$\frac{1}{T} \sum_{k=0}^{T-1} \mathbb{E}\|\nabla F(\Theta_k)\|_F^2 \le \frac{4\Delta_F}{c\kappa_c \eta T} + \frac{4r\sigma_F^2}{c\kappa_c^2 B}. \tag{40}$$

For constant stepsize and batch size, equation 40 consists of an $O(T^{-1})$ optimization term and an $O(B^{-1})$ stochastic noise floor.

B.2 Preliminary Inequalities

Lemma 1 (Properties of the nuclear-weighted polar map)**.** *Let $r = \min(m, n)$ and $\mathcal{H}(A) = \frac{1}{2}\|A\|_*^2$. The map $\mathcal{D}$ satisfies*

$$\begin{aligned} \langle A, \mathcal{D}(A)\rangle &= \|A\|_*^2, \\ \|\mathcal{D}(A)\|_2 &= \|A\|_*, \\ \|\mathcal{D}(A)\|_F^2 &= \operatorname{rank}(A)\|A\|_*^2 \le r\|A\|_*^2, \\ \mathcal{D}(A) &\in \partial\mathcal{H}(A). \end{aligned} \tag{41}$$

All identities remain valid at $A = 0$.

Proof. For $A \ne 0$, write its compact SVD as $A = U_q \Sigma_q V_q^\top$. Then

$$\langle A, U_q V_q^\top\rangle = \|A\|_*, \qquad \|U_q V_q^\top\|_2 = 1, \qquad \|U_q V_q^\top\|_F^2 = q. \tag{42}$$

This proves the norm and inner-product identities. For any matrix B, spectral–nuclear duality gives

$$\langle \mathcal{D}(A), B\rangle \le \|A\|_* \|B\|_* \le \tfrac{1}{2}\|A\|_*^2 + \tfrac{1}{2}\|B\|_*^2. \tag{43}$$

Rearranging and using $\langle A, \mathcal{D}(A)\rangle = \|A\|_*^2$ yields

$$\mathcal{H}(B) \ge \mathcal{H}(A) + \langle \mathcal{D}(A), B - A\rangle. \tag{44}$$

This is the subgradient inequality. The zero case follows from $\mathcal{D}(0) = 0$ and $\mathcal{H}(B) \ge 0$. □

Lemma 2 (Coercive quadratic bounds)**.** *For* $0 < c < 4$*, let*

$$\kappa_c = \frac{4-c}{4(1+2c)}$$

and

$$q_c(a,b) = a^2 - cab + cb^2.$$

For all $a, b \geq 0$*,*

$$q_c(a,b) \geq c\kappa_c(a+b)^2, \qquad q_c(a,b) \geq \kappa_c(a+cb)^2. \tag{45}$$

Proof. Both inequalities follow from completing the square:

$$\begin{aligned} q_c(a,b) - c\kappa_c(a+b)^2 &= \frac{((c+2)a - 3cb)^2}{4(1+2c)}, \\ q_c(a,b) - \kappa_c(a+cb)^2 &= \frac{c(3a-(c+2)b)^2}{4(1+2c)}. \end{aligned} \tag{46}$$

□

B.3 Proof of Theorem 1

Proof. All inner products below are Frobenius inner products. Nuclear and spectral norms appear only as proof tools for the unchanged exact update.

All expectations below are finite at each finite iteration. Indeed, Lemma 1 and $\|A\|_* \leq \sqrt{r}\|A\|_F$ give

$$\|\mathcal{D}(A)\|_F \leq r\|A\|_F.$$

The Lipschitz gradient has at most linear growth, and the conditional variance bound then gives finite second moments of G_k whenever Θ_k has a finite second moment. Starting from fixed Θ_0 and $M_{-1} = 0$, the momentum and parameter recursions therefore imply finite second moments of all states by induction. Smoothness and lower boundedness also make $F(\Theta_k)$ integrable.

Write

$$\begin{aligned} &H_k := \nabla F(\Theta_k), \quad \zeta_k := G_k - H_k, \qquad \tau := \frac{\beta}{1-\beta}, \\ &a_k := \|M_k\|_*, \qquad b_k := \|E_k\|_*, \qquad Q_k := a_k^2 - ca_kb_k + cb_k^2. \end{aligned} \tag{47}$$

The momentum recursion implies the exact identity

$$E_k = \tau(M_k - M_{k-1}). \tag{48}$$

Since $\mathcal{D}(M_k) \in \partial\mathcal{H}(M_k)$,

$$\begin{aligned} \langle E_k, \mathcal{D}(M_k)\rangle &= \tau\langle M_k - M_{k-1}, \mathcal{D}(M_k)\rangle \\ &\geq \tau\big(\mathcal{H}(M_k) - \mathcal{H}(M_{k-1})\big). \end{aligned} \tag{49}$$

Using $G_k = M_k + E_k$ and $\langle M_k, \mathcal{D}(E_k)\rangle \geq -a_kb_k$, we obtain

$$\begin{aligned} \langle G_k, \mathcal{U}_k\rangle &= a_k^2 + cb_k^2 + \langle E_k, \mathcal{D}(M_k)\rangle + c\langle M_k, \mathcal{D}(E_k)\rangle \\ &\geq Q_k + \tau\big(\mathcal{H}(M_k) - \mathcal{H}(M_{k-1})\big). \end{aligned} \tag{50}$$

Lemmas 1 and 2 give

$$\|\mathcal{U}_k\|_F^2 \leq r(a_k + cb_k)^2 \leq \frac{r}{\kappa_c}Q_k. \tag{51}$$

Also,

$$\|G_k\|_F \leq \|G_k\|_* \leq a_k + b_k,$$

hence

$$Q_k \geq c\kappa_c\|G_k\|_F^2. \tag{52}$$

Set $\eta_{-1} = \eta_0$ and define the nonnegative energy

$$\mathcal{V}_k := F(\Theta_k) - F_{\inf} + \tau\eta_{k-1}\mathcal{H}(M_{k-1}). \tag{53}$$

Frobenius smoothness and $H_k = G_k - \zeta_k$ imply

$$\begin{aligned} F(\Theta_{k+1}) &\le F(\Theta_k) - \eta_k\langle H_k, \mathcal{U}_k\rangle + \frac{L_F\eta_k^2}{2}\|\mathcal{U}_k\|_F^2 \\ &\le F(\Theta_k) - \tau\eta_k\big(\mathcal{H}(M_k) - \mathcal{H}(M_{k-1})\big) \\ &\quad - \eta_k\left(1 - \frac{rL_F\eta_k}{2\kappa_c}\right)Q_k + \eta_k\langle\zeta_k, \mathcal{U}_k\rangle. \end{aligned} \tag{54}$$

Because $\eta_k \le \kappa_c/(rL_F)$ and $\eta_k \le \eta_{k-1}$, adding the new momentum energy gives

$$\mathcal{V}_{k+1} \le \mathcal{V}_k - \frac{\eta_k}{2}Q_k + \eta_k\langle\zeta_k, \mathcal{U}_k\rangle. \tag{55}$$

By Frobenius Cauchy–Schwarz and Young's inequality,

$$\begin{aligned} \langle\zeta_k, \mathcal{U}_k\rangle &\le \|\zeta_k\|_F\sqrt{\frac{rQ_k}{\kappa_c}} \\ &\le \frac{1}{4}Q_k + \frac{r}{\kappa_c}\|\zeta_k\|_F^2. \end{aligned} \tag{56}$$

Therefore,

$$\mathcal{V}_{k+1} \le \mathcal{V}_k - \frac{\eta_k}{4}Q_k + \frac{r\eta_k}{\kappa_c}\|\zeta_k\|_F^2. \tag{57}$$

Unbiasedness in the Frobenius inner-product space gives

$$\mathbb{E}\left[\|G_k\|_F^2 \mid \mathcal{F}_k\right] = \|H_k\|_F^2 + \mathbb{E}\left[\|\zeta_k\|_F^2 \mid \mathcal{F}_k\right]. \tag{58}$$

Combining this identity with equation 52, we have

$$\mathbb{E}\left[Q_k \mid \mathcal{F}_k\right] \ge c\kappa_c\|H_k\|_F^2. \tag{59}$$

Taking conditional expectations in equation 57 yields

$$\mathbb{E}\left[\mathcal{V}_{k+1} \mid \mathcal{F}_k\right] \le \mathcal{V}_k - \frac{c\kappa_c\eta_k}{4}\|H_k\|_F^2 + \frac{r\eta_k\sigma_F^2}{\kappa_c B_k}. \tag{60}$$

Taking total expectations and summing over $k = 0, \dots, T-1$, and using $M_{-1} = 0$, $\mathcal{V}_0 = \Delta_F$, and $\mathcal{V}_T \ge 0$, gives

$$\frac{c\kappa_c}{4}\sum_{k=0}^{T-1}\eta_k\mathbb{E}\|\nabla F(\Theta_k)\|_F^2 \le \Delta_F + \frac{r\sigma_F^2}{\kappa_c}\sum_{k=0}^{T-1}\frac{\eta_k}{B_k}. \tag{61}$$

Dividing by $c\kappa_c S_T/4$ proves equation 39. Setting $\eta_k \equiv \eta$ and $B_k \equiv B$ gives equation 40.

Importantly, $\mathcal{U}_k$ depends nonlinearly on G_k. The proof therefore does not set

$$\mathbb{E}\left[\langle\zeta_k, \mathcal{U}_k\rangle \mid \mathcal{F}_k\right]$$

to zero; instead, this term is bounded explicitly before taking expectations. □

C Additional Experimental Details

Training protocol. Both GPT-2-small and GPT-2-medium use a sequence length of 1024 and a global batch size of 512 sequences. The training corpus is deterministically packed into contiguous sequences, with the first $4{,}096$ packed sequences reserved as a fixed validation set and the remaining sequences used for training. All experiments use bfloat16 precision and gradient clipping at 1.0. Each main comparison consists of one run per method with seed 42. Within each model setting, all methods use the same initialization, data ordering, batch configuration, validation set, and learning-rate schedule.

Parameter grouping. The compared matrix optimizer is applied to eligible two-dimensional weight matrices in the Transformer hidden layers. Fused query, key, and value projections are treated as a single matrix. Embeddings, normalization parameters, biases, the language-modeling head, and all remaining parameters are optimized using AdamW. The AdamW fallback follows the same base learning-rate schedule as the matrix optimizer at every step, without matrix-specific dimension scaling.

Matrix-optimizer implementation. AIMuon approximates the momentum and residual matrix-sign directions using five Newton–Schulz iterations:

$$D_k^M = \mathrm{NS}_5(M_k), \qquad D_k^E = \mathrm{NS}_5(E_k).$$

The Muon baseline uses $\beta = 0.95$, following the Keller Jordan / modded-nanoGPT Newton–Schulz construction together with the later LLM-style RMS scaling convention.

For each Newton–Schulz call, the input is cast to bfloat16 and transposed when the number of rows exceeds the number of columns. It is normalized by the larger of its Frobenius norm and the NS-specific stabilizer $\epsilon_{\mathrm{NS}} = 10^{-7}$. We then apply five iterations with polynomial coefficients $(3.4445, -4.7750, 2.0315)$ and transpose the output back when needed. An exactly zero input remains zero throughout this procedure.

AIMuon's nuclear-norm estimates are computed using the absolute inner products in equation 13. For clarity, its matrix-wise residual coefficient and unnormalized combined direction are

$$\lambda_k = c\frac{\widehat{\nu}_k^E}{\widehat{\nu}_k^M + \varepsilon}, \tag{62}$$

and

$$\widetilde{O}_k = \widehat{\nu}_k^M \left(D_k^M + \lambda_k D_k^E\right), \tag{63}$$

respectively. The final RMS matching follows Algorithm 1. We use $\varepsilon = 10^{-12}$ for AIMuon, distinct from both the Newton–Schulz stabilizer and the AdamW epsilon. MONA is implemented according to its proposed update with the hyperparameters specified below.

Learning-rate selection. Learning rates are selected independently for each optimizer using short pretraining runs, with the candidate achieving the lowest terminal validation loss selected for the full run. For GPT-2-small, we search over

$$\left\{5 \times 10^{-4}, 7.5 \times 10^{-4}, 1 \times 10^{-3}, 1.5 \times 10^{-3}, 2 \times 10^{-3}\right\},$$

using approximately 0.248B training tokens per candidate. For GPT-2-medium, we search over

$$\left\{5 \times 10^{-4}, 7.5 \times 10^{-4}, 1 \times 10^{-3}\right\},$$

using approximately 0.355B training tokens per candidate.

The search selects 2×10^{-3} for all compared methods on GPT-2-small and 1×10^{-3} for all compared methods on GPT-2-medium. Both selected values lie at the upper endpoints of their respective search grids; larger learning rates were not evaluated. After selection, each optimizer is restarted from the same initial parameters and trained for the full token budget.

Optimizer hyperparameters. AIMuon and Muon use the momentum coefficient $\beta = 0.95$. For AIMuon, the residual coefficient is fixed to $c = 0.05 = 1 - \beta$ in the main experiments, motivated by the Nesterov-style reference relation discussed in Sec. 3.1. The GPT-2-small sensitivity sweep over c is conducted separately after fixing this main configuration.

For MONA, we search over $\beta \in \{0.9, 0.95\}$ and select $\beta = 0.9$ according to terminal validation loss. We further use $\beta_a = 0.99$ and $\alpha = -1/[2(1 - \beta_a)] = -50$. All matrix-based optimizers use weight decay 0.1.

Parameters assigned to the AdamW fallback use $(\beta_1, \beta_2) = (0.9, 0.95)$ and $\epsilon_{\mathrm{Adam}} = 10^{-8}$. AdamW weight decay is 0.01 for parameters in the decay group and zero for normalization parameters, biases, and other no-decay parameters.

Learning-rate schedule and evaluation protocol. Both the search stage and full training use linear warmup over the first 1% of optimization steps, followed immediately by cosine decay to zero over the remaining 99%. The full runs contain 4731 optimization steps for GPT-2-small and 6772 steps for GPT-2-medium. Validation loss is evaluated every 125 steps and once again at the end of training.

In addition to final validation loss, we measure optimization efficiency using Muon's terminal unsmoothed validation loss as a fixed target. The targets are 3.2233 for GPT-2-small and 3.2026 for GPT-2-medium. AIMuon first reaches or improves upon these values at evaluated steps 4125 and 6250, respectively. The reported step reductions are computed from these discrete first-crossing checkpoints relative to Muon's full training budget, without interpolation between validation evaluations.

These crossings are measured from runs trained with the original full-budget cosine schedule. Retraining with a schedule that terminates at the crossing step would constitute a different training-budget experiment and is therefore not used for the reported comparison.

Hardware and wall-clock measurement. GPT-2-small experiments are run on four NVIDIA A100 GPUs, whereas GPT-2-medium experiments are run on four NVIDIA RTX 5090 GPUs. Figure 2 reports GPT-2-medium validation loss against elapsed training time and therefore includes the additional computation required by AIMuon's residual matrix-sign transformation. Per-step runtime, throughput, peak GPU memory, and communication overhead are not reported separately. Accordingly, the main 12.8% and 7.7% efficiency improvements refer to reductions in optimization steps rather than wall-clock time.

Additional Ablation Details. The residual-construction ablations compare three alternative ways of incorporating the gradient–momentum residual while retaining the final RMS matching. With M_k denoting the momentum and $E_k = G_k - M_k$, their directions before final RMS matching are

$$\begin{aligned} \text{Fixed residual weight:} \quad & D_k = \mathrm{NS}_5(M_k) + \lambda_{\mathrm{fix}}\,\mathrm{NS}_5(E_k), \\ \text{Inside MSign:} \quad & D_k = \mathrm{NS}_5(M_k + cE_k), \\ \text{Linear residual:} \quad & D_k = \mathrm{NS}_5(M_k) + cE_k. \end{aligned}$$

All Newton–Schulz calls follow the implementation described above.

The fixed-weight variant uses $\lambda_{\mathrm{fix}} = 0.35$, chosen to approximately match the mean effective coefficient near the middle of the full AIMuon run. This configuration is labeled Fixed $\lambda = 0.35$ in Figure 3 and Table 1. The inside-MSign and linear-residual variants use $c = 0.05$. The latter adds the raw residual E_k without separate residual normalization. These three variants use the final rescaling

$$O_k = 0.2\sqrt{\max(m,n)}\,\frac{\|\,\mathrm{NS}_5(M_k)\|_F}{\|D_k\|_F + \varepsilon} D_k.$$

For the inside-MSign variant, $c = 0.05 = 1 - \beta$ with $\beta = 0.95$, so

$$M_k + cE_k = \beta M_k + (1-\beta)G_k,$$

giving the Nesterov-style combination discussed in Sec. 3.1 before the matrix-sign transformation and final rescaling.

The no-residual variant is obtained by setting $c = 0$ in Algorithm 1, with all other operations and hyperparameters unchanged.

The normalization-removal variant retains the dimension factor $0.2\sqrt{\max(m,n)}$ but removes the final Frobenius-norm ratio from Algorithm 1. Its terminal validation loss is 3.5072.

All ablation variants use the corresponding AIMuon learning rate without independent retuning.

D AIM Framework and Parameter-Subproblem Refinement

The inherited AIM framework (Ren et al., 2026) separates a tentative geometric step, parameter refinement and a multiplier update. Gradient-difference methods such as Adan (Xie et al., 2024) and

ALTO (Zhao et al., 2025) concern momentum construction; AIMuon instead develops the spectral parameter refinement. Introducing an auxiliary variable Y, we rewrite the stochastic problem as

$$\min_{\Theta,Y} F(\Theta) \qquad \text{s.t.} \qquad Y - \Theta = 0. \tag{64}$$

Let M be the multiplier associated with the consensus constraint. For a convex penalty ψ, AIM considers

$$\mathcal{L}_\rho(\Theta, Y, M) = F(\Theta) + \langle M, Y - \Theta\rangle + \frac{\rho}{2}\psi(Y - \Theta). \tag{65}$$

AIM alternates between auxiliary-variable, parameter, and multiplier updates:

$$\begin{aligned} Y_{k+1} &\in \arg\min_{Y} \left\{ \langle M_k, Y - \Theta_k\rangle + \frac{\rho_{1,k}}{2}\psi(Y - \Theta_k) \right\}, \\ \Theta_{k+1} &\in \arg\min_{\Theta} \left\{ F(\Theta) + \langle M_k, Y_{k+1} - \Theta\rangle + \frac{\rho_{2,k}}{2}\psi(Y_{k+1} - \Theta) \right\}, \\ M_{k+1} &= M_k + (1-\beta)\frac{\rho_{2,k}}{2} D_{k+1}, \end{aligned} \tag{66}$$

where $D_{k+1} \in \partial\psi(Y_{k+1} - \Theta_{k+1})$.

When the parameter subproblem is solved exactly, choose D_{k+1} to satisfy its first-order optimality condition. Substituting this subgradient into the relaxed multiplier update yields the familiar EMA form

$$M_{k+1} = \beta M_k + (1-\beta)\nabla F(\Theta_{k+1}). \tag{67}$$

AIM therefore separates three aspects of optimizer design: the multiplier dynamics determine the momentum construction, the penalty ψ determines the update geometry, and the approximation of the parameter subproblem determines the correction applied to the tentative point.

Under Euclidean geometry, the two approximation strategies recover familiar momentum forms under the corresponding parameter choices:

$$\begin{aligned} \text{direct assignment} &\longrightarrow \text{SGD with momentum}, \\ \text{one-step fixed-point refinement} &\longrightarrow \text{Nesterov momentum}. \end{aligned} \tag{68}$$

Thus, acceleration can arise from refining the parameter subproblem rather than solely from modifying the momentum recursion. The splitting residual $Y_{k+1} - \Theta_{k+1}$ is a discrepancy between primal variables. The local spectral derivation in Sec. 3.2 instead produces the gradient–momentum residual $G_k - M_k$ as the linear coefficient of the parameter displacement. The framework motivates this construction.

We retain the EMA momentum recursion motivated by the AIM multiplier update and index it so that M_k incorporates the stochastic gradient evaluated at Θ_k.