AuON: A Survey For Linear-time Orthogonal Optimizer

Dipan Maity
dipanai.xyz@gmail.com

September 21, 2025

Abstract

Orthogonal gradient updates have recently been proposed as a promising direction for optimization in machine learning, but traditional approaches such as SVD/QR decomposition incur prohibitive computational costs $O(n^3)$ and underperform due to applying momentum after the strict orthogonalization, compared to well-tuned SGD-Momentum. Recent advances like Muon improve efficiency by moving the momentum to before the orthogonalization and producing semi-orthogonal matrices through Newton-Schulz iterations, reducing complexity to $O(n^2)$, but quadratic costs remain a bottleneck. In this work, we study on semi-orthogonal properties of updates with momentum and find out a way how to bound the momentum updates under spectral-norm trust-region and preserve the direction information without the need for semi-orthogonalization.

We propose AuON(Alternative Unit-norm momentum-updates by Normalized nonlinear scaling), a linear-time optimizer that achieves remarkable performance at linear time without producing semi-orthogonal matrices while preserving structure to guide better-aligned progress and recondition ill-posed updates. Our approach combines hyperbolic cosine RMS scaling transformations with normalization, demonstrating both effectiveness and computational efficiency compared to Newton-Schulz methods We also proposed a hybrid method(hybired-AuON) that use one iteration of Newton–Schulz Algorithm. Experiments across vision and language benchmarks demonstrate that AuON and its hybrid variant can achieve comparable performance in state-of-the-art architectures. Code will be available at `https://github.com/ryyzn9/AuON`

1 Introduction

"if you want to achieve extraordinary progress in AI, you should enhance the optimizer, as it fundamentally determines how models learn. "
Optimization in deep neural networks remains a primary challenge, particularly due to the ill-conditioning of gradient and momentum updates. Empirically, these updates often exhibit a high condition number, with most of the energy concentrated in a few dominant directions. In practical terms, the update vectors are nearly low-rank: a handful of directions dictate the optimization trajectory while many potentially informative directions may be suppressed. This imbalance is reminds us of a squashed ball that can only roll efficiently along a single axis, ignoring other pathways that may be equally important for

generalization and representation learning. To address this issue, one solution is to make all the update direction unit length; recent work has proposed orthogonalization of gradients and momentum updates to achieve the unit length. By orthogonalizing an update

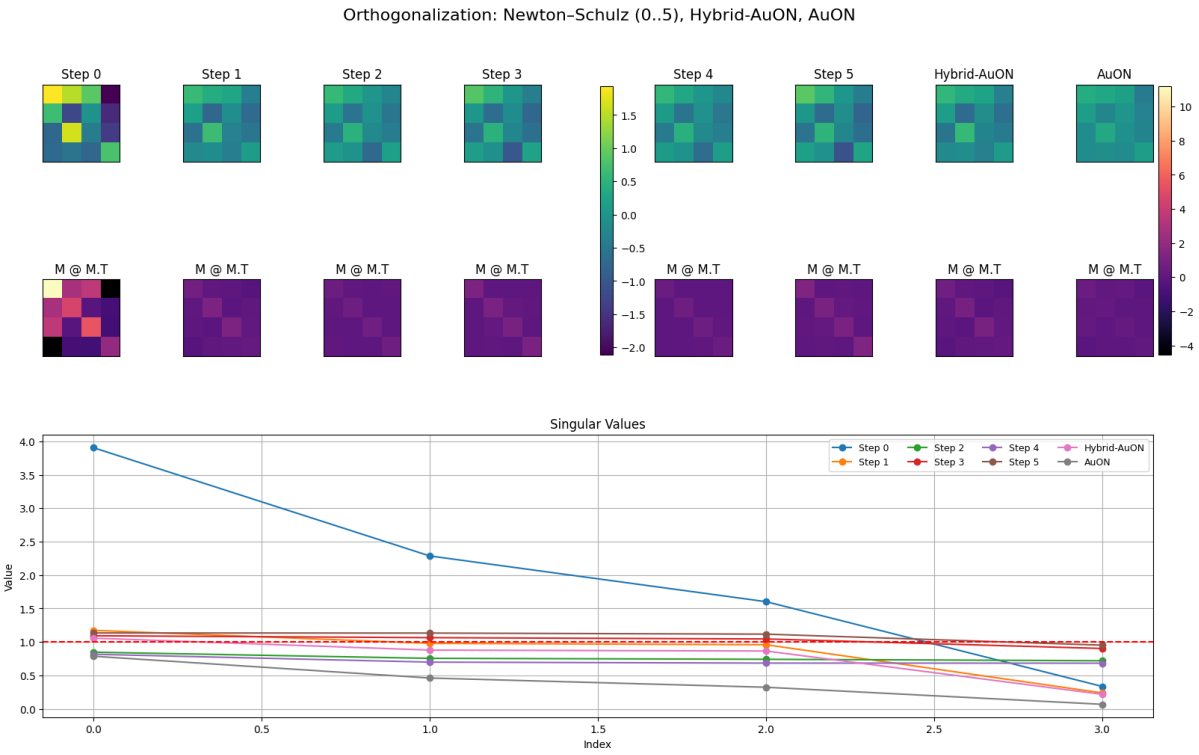

Figure 1: Visualization of the Newton–Schultz process (0.5) over 5 iterations, compared with AuON and Hybrid-AuON. The heatmaps (top) show progressive orthogonalization, with $MM^\top$ converging from a scattered structure (Step 0) to an identity-like diagonal (Step 5). The singular value plot (bottom) illustrates rapid convergence toward 1.0, confirming orthogonalization.

matrix, we effectively discard the scaling information encoded in the singular values and modify each direction to enforce perpendicularity, putting all the update length into unit vectors(unit length) but different directions. In this sense, the resulting update behaves as a "unit-norm" update in the spectral domain, emphasizing the geometric structure of the optimization landscape rather than the raw magnitude of the gradients. In simple words, Orthogonalization effectively increases the scale of other "rare directions" which have small magnitude in the update but are nevertheless important for learning. Such a perspective highlights how orthogonalization can prioritize exploration across all relevant directions, potentially mitigating the dominance of a few high-energy directions and facilitating more balanced learning dynamics (Zhang et al. 2025). Orthogonalized updates can be interpreted as spectral descent directions, which helped to ensure updates explore all directions evenly, which is crucial for generalization and representation learning.

(Tuddenham et al. 2022) proposed an approach for neural network optimization in which the gradient is first orthogonalized via singular value decomposition (SVD), followed by the application of momentum, and then the resulting momentum term is used as the update. They refer to this method as Orthogonal-SGDM. In their experiments, they observed that, even in their best-performing configuration, Orthogonal-SGDM was outperformed by a well-tuned standard SGD with momentum due to applying momentum

after strict orthogonalization, as Orthogonal-SGDM essentially damages the momentum mechanism by orthogonalizing gradients before momentum accumulation, preventing momentum from effectively reducing variance and maintaining beneficial directional information and strict orthogonality thereby erases singular-value magnitudes and overconstrains the step, meaning it over-normalizes the update by collapsing its singular-value structure to an isometry, effectively turning the step into a spectral-norm–constrained, normalized move that discards useful magnitude information. This wipes out all correlations between update directions. Makeing all the updates into unit length may be problematic as it may increase harmful alignment . Recent advances, as such as Muon (Jordan et al. 2024), improve efficiency and performance by producing a semi-orthogonal matrix using Newton–Schulz iterations rather than a full orthogonal matrix using SVD and reordering the momentum update before the semi-orthogonalization, thereby reducing the complexity to $O(n^2)$. But the computation cost of Newton-Schulz is still $O(n^2)$.

In this paper, we focus on developing an alternative approach to bound the updates with a high condition number under unit norm . Our goal is -how to achieve an impressive performance at O(n) time complexity, meaning without compromising efficiency and speed. We empirically find that normalization followed by a hyperbolic function(cosh) scale magnitude yields promising results

2 Preliminaries

2.1 orthogonalization

By orthogonalizing an update matrix $G \in \mathbb{R}^{m\times n}$ with singular value decomposition

$$G = U\Sigma V^\top,$$

The update is replaced by its orthogonal polar factor

$$Q := UV^\top.$$

This satisfies

$$Q^\top Q = I_n \quad \text{when } m \geq n, \qquad QQ^\top = I_m \quad \text{when } m \leq n,$$

thereby discarding the scaling information carried by the singular values Σ while preserving the directional subspaces encoded by the left and right singular vectors U and V.

In this sense, the resulting update behaves as unit-norm in the spectral domain—

$$\|Q\|_2 = 1$$

with a flat singular spectrum—emphasizing the geometric structure of the optimization landscape rather than the raw gradient magnitudes.

Intuitively, this equalizes per-direction gain: directions that originally had small singular values ("rare directions") are relatively amplified while dominant directions are relatively attenuated, promoting exploration across all relevant directions and mitigating the dominance of a few high-energy modes.

In practice, orientation and step size can be decoupled by using

$$\alpha Q, \qquad \alpha = \frac{\|G\|_F}{\sqrt{\text{rank}(G)}},$$

so that scale is controlled externally while orthogonalization enforces well-conditioned, balanced updates—yielding more stable and equitable learning dynamics compared to conventional gradient-descent steps.

2.2 semi-orthogonalization

Given $G \in \mathbb{R}^{m\times n}$ with singular value decomposition

$$G = U\Sigma V^\top,$$

strict orthogonalization replaces G by its polar/Stiefel projection

$$Q := UV^\top,$$

collapsing the singular spectrum to $\sigma_i(Q) = 1$ on the update subspace and making Q an isometry with

$$\|Q\|_2 = 1, \qquad Q^\top Q = I_n \;\; (\text{or} \;\; QQ^\top = I_m),$$

i.e., the Frobenius-nearest semi-orthogonal matrix that removes the amplitude information in Σ.(Higham 2000)

In the singular basis,

$$G^\top G = V\Sigma^2 V^\top$$

becomes

$$Q^\top Q = I, \qquad QQ^\top = UIU^\top = \Pi_{\mathrm{col}(G)},$$

turning the step into a spectral-norm–bounded move that can discard curvature-aligned anisotropy.

Geometrically, for Muon's RMS-to-RMS operator norm, we have

$$Q \in \arg\max_{\|X\|_{\mathrm{RMS}\to\mathrm{RMS}}\le 1} \langle X, G\rangle,$$

which is the linear minimization oracle (LMO) of a conditional-gradient step. Hence the singular values are flattened; by contrast, on the standard spectral-norm ball, the LMO yields the rank-1 solution $u_1 v_1^\top$.(Lee 2021)

To avoid overconstraint, semi-orthogonal schemes such as Muon orthogonalize only the momentum M_t to

$$Q_t = \mathrm{polar}(M_t),$$

and decouple scale via an RMS-to-RMS factor α, giving

$$W_{t+1} = (1 - \eta_t\lambda)W_t + \eta_t\,\alpha\,Q_t.$$

In practice, $Q_t Q_t^\top$ is computed efficiently via a low-order Newton–Schulz iteration, and α is chosen to match update RMS across shapes, enabling stability and learning-rate transfer. Semi-orthogonalization stabilizes training by bounding spectral energy and equalizing directional gains, preventing overshoot along sharp curvature, reducing oscillations, and enabling larger learning rates by decoupling orientation from scale(Liu et al. 2025)

2.3 Orthogonalized Momentum as a Spectral Trust-Region Method

Recent advances demonstrate that orthogonalized momentum in deep learning optimizers, particularly the Muon optimizer, admits a principled interpretation as the solution to a non-Euclidean trust-region subproblem under the spectral norm constraint (Kovalev 2025). The core update rule can be formulated as

$$X_{k+1} = X_k - \eta O_k, \quad O_k = \mathrm{Orth}(\nabla F(X_k)),$$

where $\mathrm{Orth}(\cdot)$ denotes the SVD-based orthogonalization operator that computes

$$M = U\Sigma V^\top \quad \Longrightarrow \quad \mathrm{Orth}(M) = UV^\top,$$

yielding the steepest descent direction under the spectral norm metric.

Momentum Integration The momentum component follows the exponential moving average

$$m_{k+1} = (1-\alpha)m_k + \alpha\, g(x_k;\xi_k),$$

where $g(x_k;\xi_k)$ represents an unbiased stochastic gradient estimate. The orthogonalized update then solves the trust-region subproblem

$$x_{k+1} = \arg\min_x \Big\{ \langle \mathrm{Orth}(m_{k+1}),\, x\rangle : \|x - x_k\|_2 \le \eta \Big\}.$$

This formulation explicitly constrains parameter updates within a trust region while ensuring the search direction maintains unit spectral norm.

Theoretical Advantages The orthogonalization-first approach provides superior variance reduction compared to alternative momentum–orthogonalization orderings. By applying orthogonalization to the momentum vector before the parameter update, the method preserves the accumulated directional information while eliminating scale-dependent instabilities. This design choice demonstrates both theoretical guarantees for convergence under non-convex objectives and empirical improvements in training stability across diverse architectures (Liu et al. 2025).

3 Methods

We hypothesize that forcing all update directions to unit length can be problematic, as not all directions contribute equally to optimization progress—some may be harmful (having a negative impact) or irrelevant to loss reduction. Our goal is to develop an alternative method that removes the harmful directions or alignments and preserves the beneficial properties of near semi-orthogonalization while selectively scaling directions under unit-norm : decrease the scales of "rare updates directions" than the dominant update directions and keep them all under a unit-norm trust region, meaning prioritizing directions with favorable conditioning that correspond to well-conditioned subspaces of the loss landscape under a spectral-norm trust-region.One solution is to apply a temperature-scaled softmax update matrix, followed by L2 renormalization, to bound the step under a trust-region. But computing softmax may be problematic as it comes with its own bottleneck, and it does not preserve the semi-orthogonal property that is needed for an optimizer like Muon . We empirically find out that the normalization with hyperbolic

functions (cosh) help us achieve spectral-norm trust-region (Kovalev 2025) and helps us preserve the near semi-orthogonal properties that is more stable and equitable learning dynamics compared to conventional gradient-descent steps, and stabilizes training by bounding spectral energy into a unit vector and equalizing directional gains, preventing overshoot along sharp curvature, reducing oscillations, and enabling larger learning rates by decoupling orientation from scale(M. A. Peletier and Schlichting 2023)

3.1 Nonlinear reshaping via hyperbolic cosine RMS scaling

Our main goal is to keep all the updated directions under unit spectral norm and remove the hermful directions. We empirically find out the updated matrix divided by the scale factor of the RMS magnitude of cosh() helps us bound the dominant update directions which have a high condition number under unit spectral norm, and helps us to preserve the near semi-orthogonal-like properties.By doing this, we remove the harmful alignment and stabilize the updates and equitable learning dynamics. The overall equation is

$$X = \frac{G}{\|G\| + 10^{-7}}$$

$$\text{update} = X$$

$$x = \cosh(\text{update})$$

$$\text{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$$

$$G = \frac{\text{update}}{\text{rms} + 10^{-8}}$$

where

$$\cosh(z) = \frac{e^z + e^{-z}}{2}$$

For large values of $|z|$, $\cosh(z)$ grows exponentially, while for small values of z,

$$\cosh(z) \approx 1 + \frac{z^2}{2}.$$

Thus, cosh magnifies meaningful deviations while remaining symmetric and smooth.

This encourages a spread of activations (diversity) without enforcing strict orthogonality (M. A. Peletier and Schlichting 2023) and X is the updated momentum vector direction

Effect of the Hyperbolic Cosine RMS Magnitude.

Define rms $:= \|\cosh(\text{update})\|_F/\sqrt{N}$, where cosh is applied *only* to compute a global, tail-sensitive scale (M. Peletier and Schlichting 2022). Because cosh is even and rapidly increasing in $|x|$, heavy tails inflate rms, which reduces the overall step size when forming $U := \text{update}/(\text{rms}+10^{-8})$. Crucially, cosh is not applied to the propagated vector: U is a uniform rescaling of update, so the signs and all relative component ratios of update are preserved in U. This yields scale invariance with tail-aware damping, without introducing per-coordinate reweighting in the final update.

Layman's terms. First, fix the raw step's size; then gauge how "spiky" it is using cosh; finally, shrink the whole step more if it looks spiky. The direction and internal proportions of the step stay the same.

Near–Semi–Orthogonality.
Exact orthogonality. A matrix $W \in \mathbb{R}^{m\times n}$ is orthogonal (semi-orthogonal if $m \neq n$) when

$$W^\top W = I_n \quad \text{or} \quad WW^\top = I_m,$$

which preserves Euclidean inner products and hence norms and angles exactly.

Method (equations). Let $G \in \mathbb{R}^{m\times n}$ be a gradient/update and $N := mn$. Define

$$\text{update} := \frac{G}{\|G\|_F + 10^{-7}}, \qquad \text{rms} := \frac{\big\|\cosh(\text{update})\big\|_F}{\sqrt{N}}, \qquad U := \frac{\text{update}}{\text{rms} + 10^{-8}}.$$

Equivalently, $U = X/(r+\varepsilon)$ with $X = \text{update}$, $r = \text{rms}$, $\varepsilon = 10^{-8}$.

Immediate implications.

- *Scale invariance.* For any $c > 0$, replacing G by cG leaves update (and thus U) unchanged up to ε-terms.
- *Tail-aware global scaling.* Heavy tails inflate rms via cosh, reducing the global magnitude of U when the update is spiky.
- *No per-component reweighting.* U is a uniform rescaling of update; signs and relative component ratios of update are preserved in U.

Norms and "balanced sphere." This construction does not enforce unit RMS for U. Indeed,

$$\|\text{update}\|_F \approx 1, \tag{1}$$

$$\|U\|_F = \frac{\|\text{update}\|_F}{\text{rms} + 10^{-8}} \approx \frac{1}{\text{rms} + 10^{-8}}, \tag{2}$$

$$\text{RMS}(U) = \frac{\|U\|_F}{\sqrt{N}} \approx \frac{1}{\sqrt{N}\,(\text{rms} + 10^{-8})} \tag{3}$$

Thus, there is no unit-L2 or unit-RMS constraint on U; the overall step length decreases as the tail-sensitive scalar rms increases.

Relation to near semi-orthogonality. Let

$$M := U^\top U \in \mathbb{R}^{n\times n}.$$

By the Frobenius–trace identity,

$$\text{tr}(M) = \|U\|_F^2,$$

and

$$\alpha := \frac{1}{n}\,\text{tr}(M),$$

which equals the average column ℓ_2 norm squared.

However, under the mapping above, $\mathrm{tr}(M)$ is determined by rms and is not generally $N = mn$ unless an extra unit-RMS rescale is applied to U.

Off-diagonal correlations

$$M_{ij} = \langle U_{:i}, U_{:j} \rangle$$

are not explicitly zeroed by this mapping, so it promotes scale invariance and approximate isotropy rather than exact semi-orthogonality.

- *Cross-correlations.* Off-diagonals of M are scaled copies of those in update$^\top$update; they are not explicitly suppressed.
- *Isotropy.* This mapping alone does not drive M toward αI_n. Achieving near semi-orthogonality typically requires an additional correlation-reducing step (e.g., per-column RMS normalization, light whitening, or a spectral penalty $\|U^\top U - \alpha I\|_F^2$ with $\alpha = \frac{1}{n}\,\mathrm{tr}(U^\top U)$). see Appendix A for more information

Practical implication.
The update is scale-invariant and tail-aware: heavy tails trigger stronger global shrinkage via rms, helping prevent blow-ups while preserving the direction and internal proportions of the step. When approximate isotropy or near semi-orthogonality is desired, pair this normalization with a lightweight correlation-suppressing operation.

3.2 Hybrid Approach

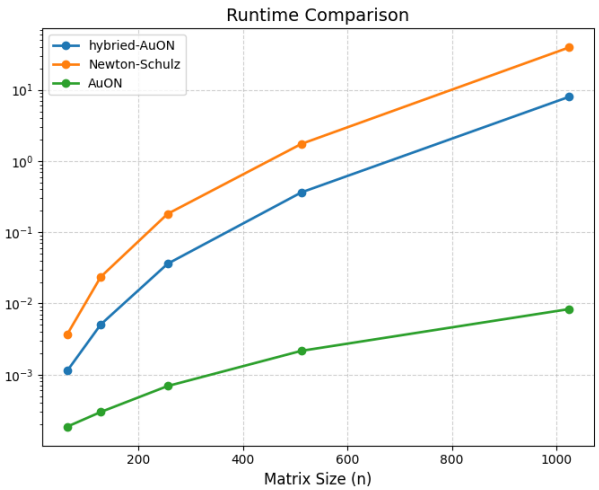

Figure 2: comparison of computation efficiency of different methods on (nxn) ramdom matrices

The hybrid approach include only one iteration of Newton-Schulz and Nonlinear re-shaping via hyperbolic cosine RMS scaling. this helps use improve the performance only using one iteration.

$$A = XX^\top$$

$$B = bA + cA^2$$

$$X \leftarrow aX + BX$$

$$G_{\text{new}} = \frac{\text{update}}{\text{rms} + \delta}, \quad \text{with update} = X, \quad \text{rms} = \frac{1}{N}\|\cosh(X)\|_F^2.$$

it helps us to achieve near semi-orthogonality and bound the updates under spectral-norm trust-region with comparable less computation than Newton-Schulz 5 times iteration in Muon and can achive impressive performance compare to Adamw and Muon

4 Experiments

4.1 Language Modeling

We evaluate our approach using 4X L4 GPU on the **SmolLM-Corpus** dataset(Ben Allal et al. 2024) , consisting of 500k tokens. The underlying model is a nanoGPT(Karpathy 2022) with FlashAttention-2(Dao 2023) rotary position embeddings (RoPE)(Su et al. 2023), RMSNorm (Huang et al. 2019), and SwiGLU activations (Shazeer 2020). For the **Small configuration**, we use a hidden size of 512, 6 layers, 8 attention heads, and a feed-forward dimension of 1536. Training is conducted for 6000 steps with a global batch size of 128. We compare AuON, AdamW (Loshchilov and Hutter 2019), Hybrid-AuON, and MuON under similar training conditions, with learning rates tuned separately: $\eta_{\text{adamw}} = 0.003$, $\eta_{\text{auon}} = 0.055$. and $\eta_{\text{muon}} = 0.01$. (Rosić and Claude 2025)

Table 1: Training Results on Tiny (Run 1). All optimizers are trained under identical settings.

Optimizer	Total Params	Opt. Params	Time (s)	Loss	Acc	PPL
AuON	40,901,120	15,728,640	1919.2	0.4305	0.8667	1.54
AdamW	40,901,120	25,172,480	1918.9	0.0686	0.9846	1.07
Hybrid-AuON	40,901,120	15,728,640	2285.4	0.0422	0.9908	1.04
MuON	40,901,120	15,728,640	2303.6	0.0375	0.9919	1.04

4.2 vision task

We evaluated **AdamW** and the proposed **Auon optimizer** on the CIFAR-10 dataset under a reduced-scale training protocol. A random seed (42) ensured reproducibility. The dataset was split into **15,000 training**, **1,500 validation**, and **5,000 test** samples. Data loading used a batch size of 32 with standard preprocessing.
Training configuration: 100 epochs, learning rate $= 1 \times 10^{-3}$, Muon LR $= 0.055$, weight decay $= 1 \times 10^{-4}$, momentum $(\beta_1, \beta_2) = (0.9, 0.99)$. The network contained 19.90M parameters.
Results (Test Accuracy):

- AdamW: 76.0%
- Auon: 73.3%
- $\Delta = -2.7$ percentage points

AdamW outperformed Auon in this small-scale setting, though performance may be affected by the limited dataset size and training duration.

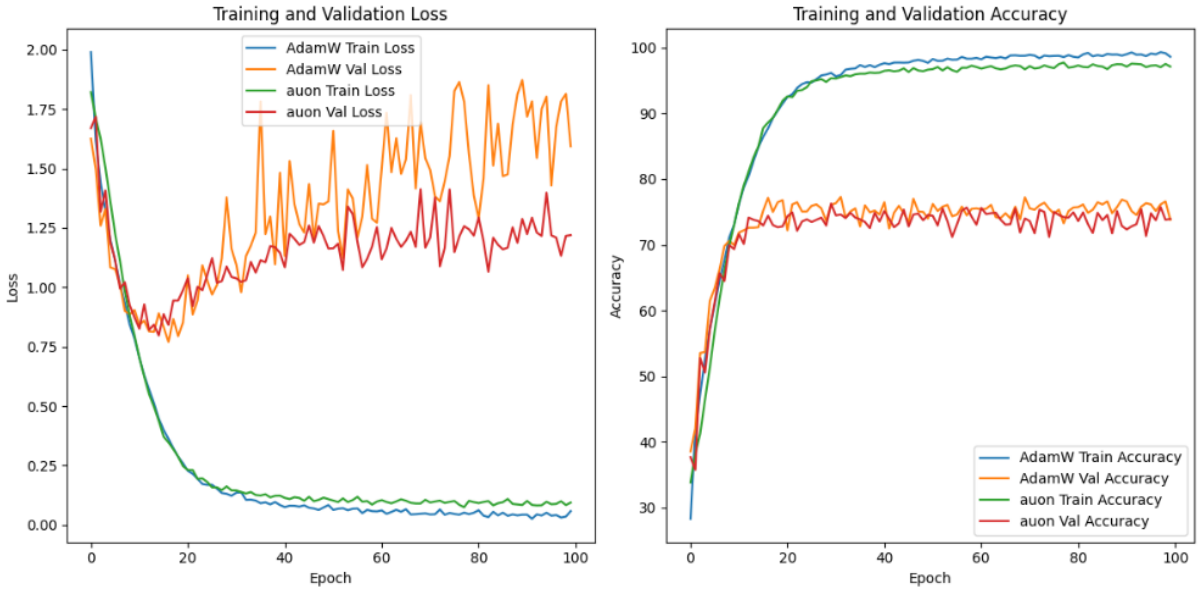

Figure 3: Training and validation curves on CIFAR-10 with AdamW and AuON optimizers. (Left) Loss decreases steadily for AdamW, while AuON exhibits higher and more unstable loss. (Right) Accuracy shows AdamW converging to ∼76% while AuON plateaus around 73%.

But increasing the batch size from 32 to 128 can improve the performance **Results (Test Accuracy):**

- AdamW: 76.32%
- Auon: 76.22%
- $\Delta = -0.10$ percentage points

5 Conclusion

In this paper, we only focus on the linear time Optimizer that has unit-spectral-norm and other semi-orthogonal properties that are needed for stabilized training without the need for the proper semi-orthogonalization. Through our experiment, we suspect that Auon and its Hybrid-variant may suffer from exploding attention logits(Team et al. 2025) on Large parameter models, due to a lack of higher GUP resources; we cannot conduct such experiments. We can use qk-clipping as describe in (Team et al. 2025) to reduce the effect. We empirically find out that increasing the model parameters increases the AuON accuracy by up to 92 percent, and furthermore, on the downstream tasks. As future work, we plan to evaluate our approach on the NanoGPT speedrun using H100 GPUs, to assess its performance in a larger-scale, practical training setting.

References

Ben Allal, Loubna, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra (July 2024). *SmolLM-Corpus*. URL: https://huggingface.co/datasets/HuggingFaceTB/smollm-corpus.

Dao, Tri (2023). *FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning*. arXiv: 2307.08691 [cs.LG]. URL: https://arxiv.org/abs/2307.08691.

Higham, Nicholas (Mar. 2000). "Matrix Nearness Problems and Applications". In: URL: https://www.researchgate.net/publication/2640282_Matrix_Nearness_Problems_and_Applications.

Huang, Shiyu, Yuxin Su, Xuezhe Ma, and Noah A. Smith (2019). "Root Mean Square Layer Normalization". In: *arXiv preprint arXiv:1910.07467*.

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

Karpathy, Andrej (2022). *NanoGPT*. https://github.com/karpathy/nanoGPT.

Kovalev, Dmitry (2025). *Understanding Gradient Orthogonalization for Deep Learning via Non-Euclidean Trust-Region Optimization*. arXiv: 2503.12645 [cs.LG]. URL: https://arxiv.org/abs/2503.12645.

Lee, James R. (2021). *Von Neumann's Inequality and Unitarily-Invariant Norms*. Lecture notes, CSE599I, Spring 2021. Instructor: James R. Lee. URL: https://example.edu/cse599i/vonneumann-notes.pdf (visited on 09/19/2025).

Liu, Jingyuan, 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 (2025). *Muon is Scalable for LLM Training*. arXiv: 2502.16982 [cs.LG]. URL: https://arxiv.org/abs/2502.16982.

Loshchilov, Ilya and Frank Hutter (2019). *Decoupled Weight Decay Regularization*. arXiv: 1711.05101 [cs.LG]. URL: https://arxiv.org/abs/1711.05101.

Peletier, Mark and André Schlichting (Aug. 2022). "Cosh gradient systems and tilting". In: *Nonlinear Analysis*, p. 113094. DOI: 10.1016/j.na.2022.113094.

Peletier, Mark A. and André Schlichting (June 2023). "Cosh gradient systems and tilting". In: *Nonlinear Analysis* 231, p. 113094. ISSN: 0362-546X. DOI: 10.1016/j.na.2022.113094. URL: http://dx.doi.org/10.1016/j.na.2022.113094.

Rosić, Vuk and Claude (2025). *Muon vs AdamW: Learning Rate And Scaling Small LLMs*. URL: https://github.com/vukrosic/muon-optimizer-research.

Shazeer, Noam (2020). *GLU Variants Improve Transformer*. arXiv: 2002.05202 [cs.LG]. URL: https://arxiv.org/abs/2002.05202.

Su, Jianlin, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu (2023). *RoFormer: Enhanced Transformer with Rotary Position Embedding*. arXiv: 2104.09864 [cs.CL]. URL: https://arxiv.org/abs/2104.09864.

Team, Kimi et al. (2025). *Kimi K2: Open Agentic Intelligence*. arXiv: 2507.20534 [cs.LG]. URL: https://arxiv.org/abs/2507.20534.

Tuddenham, Mark, Adam Prügel-Bennett, and Jonathan Hare (2022). *Orthogonalising gradients to speed up neural network optimisation*. arXiv: 2202.07052 [cs.LG]. URL: https://arxiv.org/abs/2202.07052.

Zhang, Minxin, Yuxuan Liu, and Hayden Schaeffer (2025). *AdaGrad Meets Muon: Adaptive Stepsizes for Orthogonal Updates*. arXiv: 2509.02981 [cs.LG]. URL: https://arxiv.org/abs/2509.02981.